Team aimoment at PAN 2026: Post-Hoc Synonym Watermarking with Dual-Key Type-Level Detection

Notebook for the PAN Lab at CLEF 2026

Momen Ibrahim[1,*], Nagwa El-Makky[1] and Marwan Torki[1]

[1]Computer and Systems Engineering Department, Faculty of Engineering, Alexandria University, Alexandria, Egypt

Abstract

Post-hoc watermarking of pre-existing human text must simultaneously achieve high TP (detection rate), near-zero FP, and semantic fidelity — a three-way tension sharpened when an automated obfuscation attack removes part of the embedded signal.

We address this problem by partitioning the English vocabulary into green and red word types using a cryptographic HMAC key, then globally replacing red-listed types with green-listed synonyms sourced from WordNet and BERT fill-mask. A key design choice is *type-level* detection: the Z-test counts unique word types rather than token occurrences, producing a null distribution that is better centred for the repetitive vocabulary of EU Parliament political speech.

The central empirical finding of this paper is a precision–robustness trade-off in dual-key AND detection. Requiring both of two independent Z-tests to pass simultaneously reduces $P(\text{FP})$ from $\approx 6.7\%$ (single key) to $\approx 0.45\%$ (dual key) — but the AND condition introduces brittleness: if the obfuscation attack suppresses either key's signal below threshold, the detection fails even if the other key survives strongly.

We submitted four systems to TIRA [1]. On the spot-check dataset, the dual-key system achieves TWF = 0.971 (BA = 0.995, TP = 0.496, TN = 0.499), the highest among all submissions. On the official PAN 2026 test set, the single-key system achieves the best TWF of **0.775** (BA = 0.904, TP = 0.468, TN = 0.032), while the dual-key system drops to TWF = 0.734 (BA = 0.753, TP = 0.254, TN = 0.499) — confirming the trade-off under the stronger official attack.

Keywords

text watermarking, synonym substitution, type-level detection, dual-key statistical watermarking, precision–robustness trade-off, PAN 2026

1. Introduction

Text watermarking embeds an invisible signal into a document that can later be detected to verify provenance, even after an adversarial obfuscation attack. The PAN@CLEF 2026 Text Watermarking task [2, 3] specifically targets *post-hoc* watermarking: the signal must be embedded into pre-existing human-authored EU Parliament speeches, not guided during generation. The evaluation metric is the *Text Watermarking Fidelity* (TWF):

$$\text{TWF} = \max(\text{BLEU}, \text{BERTScore}) \times \text{BA}, \tag{1}$$

where BLEU and BERTScore measure how much the watermarking changes the text, and Balanced Accuracy BA penalises both missed detections and false alarms equally; TP (True Positive) is the fraction of watermarked texts correctly detected and TN (True Negative) is the fraction of originals correctly rejected.

Our approach adapts the green-list vocabulary partitioning of Kirchenbauer et al. [4] and the post-hoc synonym substitution of PostMark [5] to this setting. We extend both with two design decisions: (i) **type-level detection**, which counts unique word types rather than token occurrences and produces

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

*Corresponding author.

es-momen.ibrahim2019@alexu.edu.eg (M. Ibrahim); nagwamakky@alexu.edu.eg (N. El-Makky); mtorki@alexu.edu.eg (M. Torki)

https://github.com/Momen-Ibrahim-Fawzy/ (M. Ibrahim)

0009-0001-7287-0705 (M. Ibrahim); 0000-0002-9571-7543 (N. El-Makky); 0000-0002-6149-1718 (M. Torki)

a better-calibrated null distribution for political speech; and (ii) **dual-key AND detection**, which requires two independent Z-tests to pass simultaneously, suppressing $P(\text{FP})$ from $\approx 6.7\%$ to $\approx 0.45\%$.

The central scientific contribution of this paper is not a claim that dual-key detection always wins. Our official test results show the opposite: the single-key system outperforms the dual-key system on the PAN 2026 test (TWF = 0.775 vs. 0.734). The contribution is an **empirical characterisation of the precision–robustness trade-off** in AND-condition detection: dual-key schemes are extremely precise but brittle under attacks that can suppress either signal independently. This trade-off has not been systematically studied for synonym-substitution watermarking and is the main finding we report.

2. Related Work

Token-level green-list watermarking. Kirchenbauer et al. [4] introduced the KGW scheme for LLM-generated text: before sampling each token, the vocabulary is partitioned using a hash of the previous token, and the model is nudged to prefer green tokens. Detection applies a one-sided Z-test on the token-level green fraction. Our work adapts this principle to post-hoc synonym substitution of *existing* human text, and replaces token-level counting with type-level counting to reduce vocabulary-bias effects in repetitive domain text.

Post-hoc synonym substitution. PostMark [5] demonstrated that substituting words with semantically similar alternatives via a masked language model yields competitive robustness while preserving meaning. Our system extends PostMark with dual-key AND detection and an empirically calibrated null hypothesis, and quantifies the precision–robustness trade-off that the dual-key design introduces.

Provably robust watermarking. Zhao et al. [6] proposed a scheme with formal robustness guarantees under edit-distance-bounded attacks. Our results complement this line of work by showing that precision improvements from multi-key AND conditions can *hurt* robustness against word-substitution attacks.

WordNet-based watermarking. Earlier systems use synonym substitution as a steganographic channel [7]. Our contribution over these baselines is the cryptographic HMAC key binding, the type-level Z-test, and the systematic study of AND-condition brittleness.

3. Methodology

3.1. Vocabulary Partition via HMAC

The core primitive is a deterministic, key-dependent binary partition of the English vocabulary. For a secret key K and a word w, we define:

$$\text{bit}(w; K) = \text{HMAC-SHA256}(K,\, w_{\text{lower}}) \bmod 2, \tag{2}$$

where w_{lower} is the lowercase form of w. Words with $\text{bit}(w; K) = 1$ are *green*; those with bit $= 0$ are *red*. Because HMAC-SHA256 is a pseudorandom function, the partition is computationally independent for any two distinct keys $K_1 \neq K_2$.

3.2. Type-Level Detection

A design choice that distinguishes our system from token-level watermarking is *type-level detection*: each unique scoreable word type is counted exactly once, regardless of how many times it appears in the text.

$$Z^{(K)}(t) = \frac{\hat{p}^{(K)} - p_0^{(K)}}{\sqrt{0.25\,/\,n}}, \tag{3}$$

where n is the number of unique scoreable word types in t and $\hat{p}^{(K)} = |\{w \in T : \text{bit}(w; K) = 1\}| \,/\, n$ is the observed green-type fraction. $p_0^{(K)}$ is an empirically calibrated null fraction (Section 3.5).

Token-level counting inflates the contribution of high-frequency political terms (e.g. *parliament*, *committee*, *member*) that appear dozens of times in a single speech; a single substitution of such a term would then dominate the Z-statistic. Type-level counting treats each word identity once, giving each substitution equal statistical weight regardless of repetition. For EU Parliament speeches, where political vocabulary is highly repetitive but type diversity is still broad, this produces a more stable null distribution.

The normal approximation $Z^{(K)} \sim \mathcal{N}(0, 1)$ under the null holds because the HMAC function assigns each vocabulary type independently to green or red with probability 0.5; for texts with $n \geq 20$ scoreable types the Central Limit Theorem provides an adequate Gaussian approximation, and for two distinct keys the Z-scores are computationally independent by the pseudorandomness of HMAC.

3.3. Dual-Key AND Detection

A text is classified as watermarked if and only if *both* Z-tests exceed the threshold:

$$Z^{(K_1)}(t) \geq Z_\tau \quad \text{AND} \quad Z^{(K_2)}(t) \geq Z_\tau, \tag{4}$$

with $Z_\tau = 1.5$. Under the null, the probability that both Z-scores exceed Z_τ by chance is:

$$P(\text{FP}) \approx P(Z_1 \geq 1.5)^2 \approx 0.067^2 \approx 0.0045, \tag{5}$$

reducing expected false positives on a 30-text corpus from ≈2 (single key) to ≈0.1 (dual key). Figure 1 shows the resulting separation in the (Z_1, Z_2) plane on the training corpus.

The AND condition and its cost. As we show empirically in Section 5, this precision advantage comes with a robustness cost. The AND condition fails if the obfuscation attack suppresses *either* key's Z-score below Z_τ, even if the other key's signal is strong. A single-key system requires only one signal to survive. This asymmetry is the fundamental precision–robustness trade-off characterised in this paper.

3.4. Watermark Embedding

Watermarking operates on five Penn Treebank part-of-speech (POS) categories [8]: $\mathcal{P} = \{$NN (singular noun), NNS (plural noun), JJ (adjective), VB (base-form verb), VBP (non-3rd-person present verb)$\}$. Embedding proceeds in three sequential stages. Figure 2 illustrates the full pipeline.

Stage 1 — Key-1 Embedding

1. **Candidate selection.** Tokenise, POS-tag, and compute $\text{bit}(w; K_1)$ for every unique type $w \in \mathcal{P}$ that has at least one WordNet synonym. Collect red types (bit $= 0$) with at least one green synonym.
2. **Frequency-first ranking.** Sort candidates by (a) types appearing ≥ 2 times before singletons, then (b) descending frequency within each group. High-frequency types are harder for an attacker to erase completely; prioritising them maximises attack robustness. Select the top $M = 12$ types; the remainder form the *amplification pool*.
3. **Global replacement.** For each selected type, replace every occurrence via word-boundary substitution, preserving capitalisation. Prefer synonyms that are green under *both* keys (*double-green*), to pre-boost the Key-2 Z-score at no additional substitution cost.
4. **Signal amplification.** If $Z^{(K_1)} < \tau_{\text{target}} = 2.5$ after step 3, continue substituting from the amplification pool until the target is reached.
5. **Semantic fallback.** If $Z^{(K_1)} < 2.5$ after step 4, query BERT fill-mask [9] with synonym-focused templates (e.g. "*X is also known as [MASK]*") for red types lacking a green WordNet synonym. Candidates are retained only if they (a) appear in WordNet and (b) have SBERT cosine similarity ≥ 0.45 with the original word using `all-mpnet-base-v2` [10].

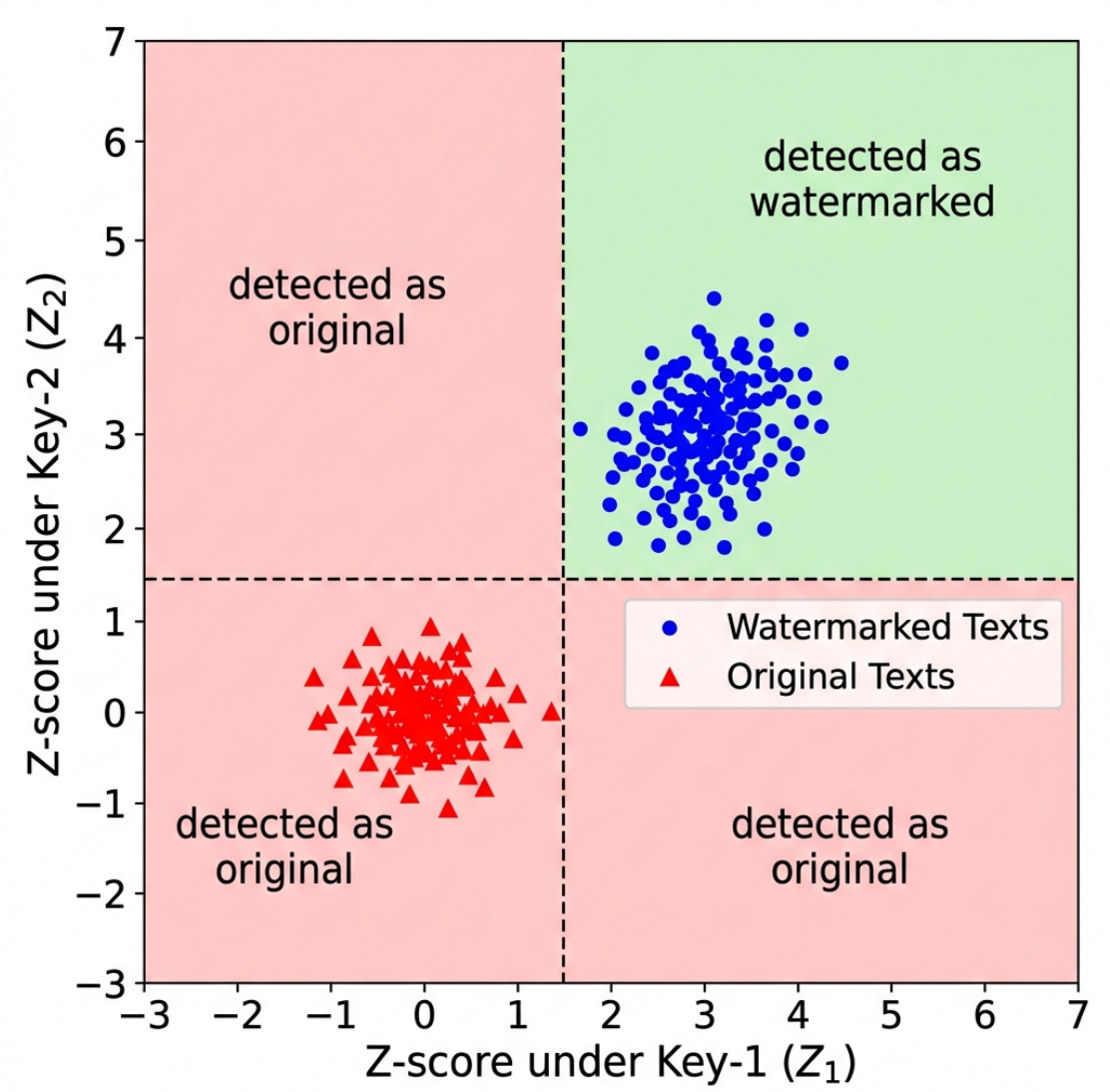

Figure 1: Z-score scatter plot for 600 training texts (300 watermarked, 300 original). Watermarked texts (blue circles) cluster at $(Z_1, Z_2) \approx (3, 3)$; originals (red triangles) scatter near the origin. Dual-key detection (green quadrant: $Z_1 \geq 1.5$ *and* $Z_2 \geq 1.5$) achieves perfect separation on all 600 texts.

Stage 2 — Key-2 Embedding

The same procedure runs for K_2, excluding types already substituted in Stage 1. Stage 2 may lower $Z^{(K_1)}$ by replacing some Key-1 green types with Key-2 green synonyms; this is addressed in Stage 3.

Stage 3 — Dual-Key Restoration

Up to three restoration rounds alternate between boosting $Z^{(K_1)}$ and $Z^{(K_2)}$ to a conservative target $\tau_{\text{corr}} = 1.8$, using the types not yet substituted by the other stage. The conservative target (rather than the full 2.5) prevents over-substitution that would exhaust the candidate pool for the partner key. Restoration terminates early when both $Z^{(K_1)} \geq Z_\tau$ and $Z^{(K_2)} \geq Z_\tau$.

3.5. Empirical Null Calibration

Equation (3) uses calibrated null fractions $p_0^{(K_1)} = 0.513$ and $p_0^{(K_2)} = 0.508$ rather than the theoretical 0.500. These are estimated by averaging the observed green-type fractions across 300 original training texts; they account for a small but consistent vocabulary bias of EU Parliament speeches under our

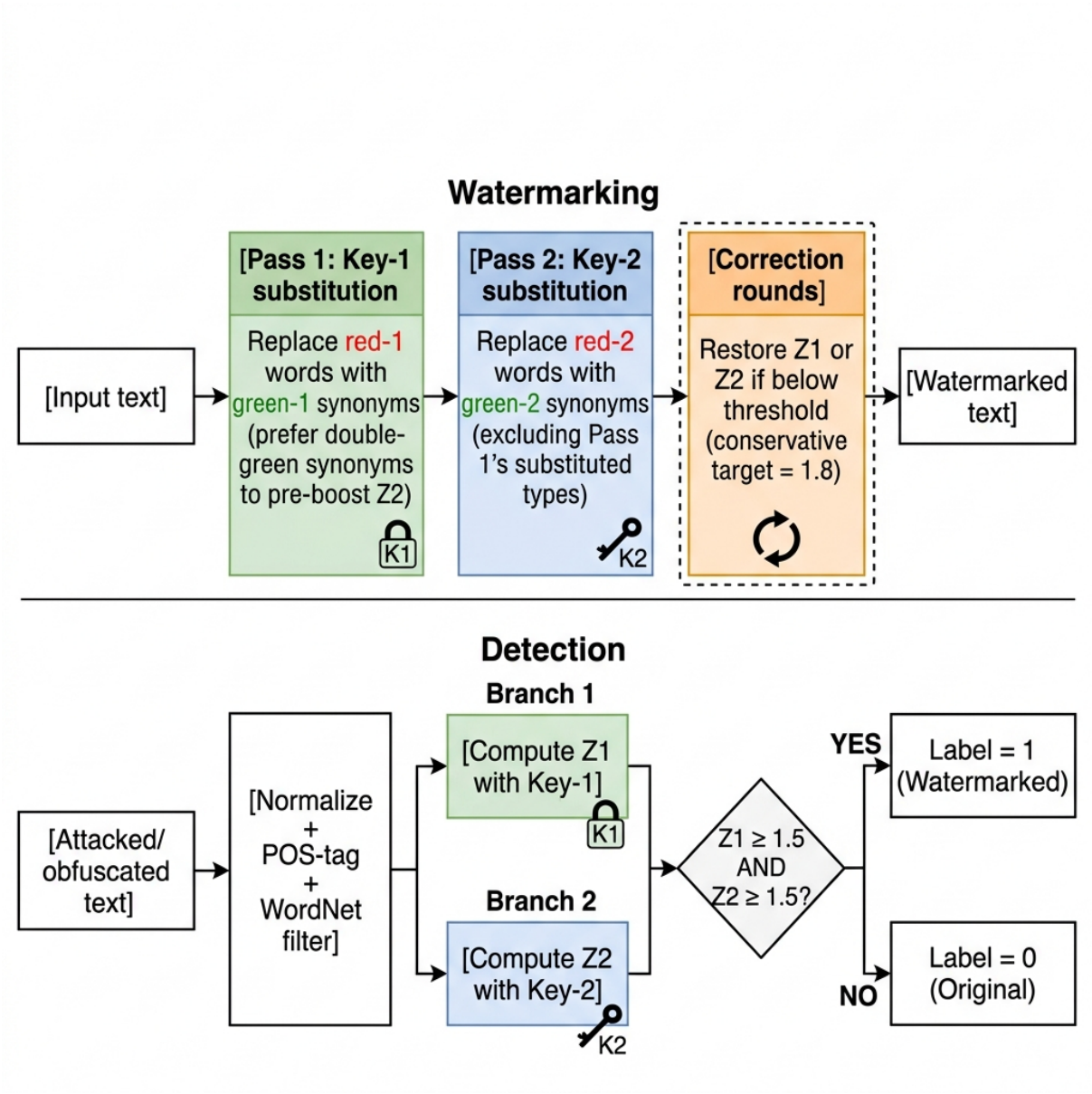

Figure 2: Dual-key watermarking and detection pipeline. Embedding applies three sequential stages (Key-1, Key-2, dual-key restoration); detection requires both Z-tests to exceed $Z_\tau = 1.5$.

specific HMAC keys. The calibrated null re-centres the Z-distribution for original texts, reducing the single-key FP from $\approx 10.3\%$ to $\approx 5.5\%$ before the dual-key multiplication.

3.6. Text Hardening

Before computing Z-scores, the detector applies NFKC unicode normalisation, homoglyph remapping (Cyrillic/Greek look-alikes to ASCII), and zero-width character removal, to defend against character-level attacks that alter the perceived word without changing its meaning.

3.7. Part-of-Speech Coverage

Experiments with only $\{\texttt{NN}, \texttt{JJ}\}$ left 28 out of 300 training texts undetectable (average 11.6 scoreable types per difficult text). Adding $\{\texttt{NNS}, \texttt{VB}, \texttt{VBP}\}$ raised the average to 24.6 scoreable types, eliminating all 28 failure cases. A POS-specific WordNet path-similarity floor filters semantically poor candidates:

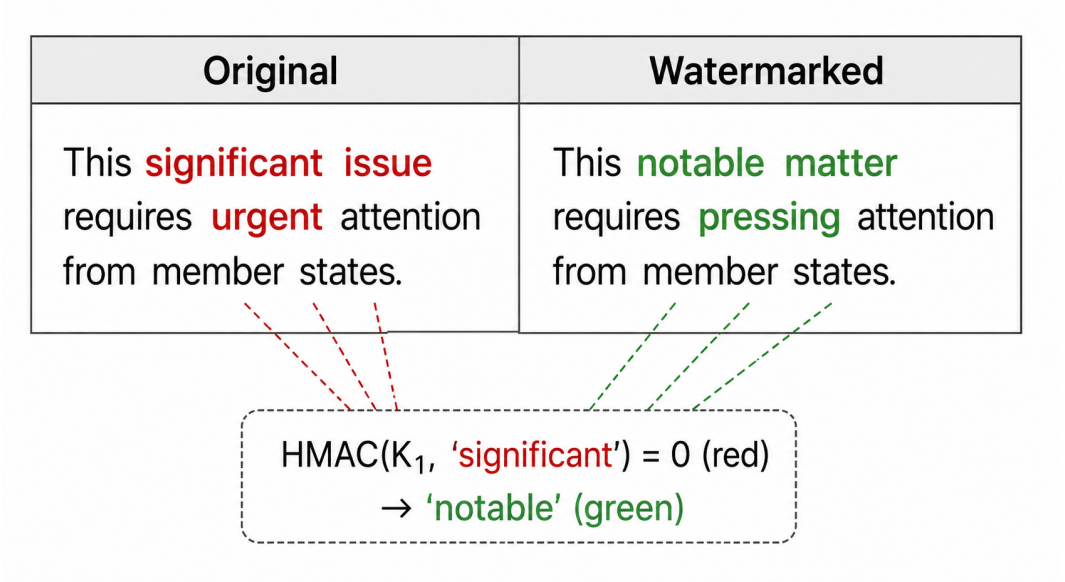

Figure 3: Example watermark substitution. Words assigned to the red list (bit$(w; K_1) = 0$) are replaced globally by semantically equivalent green-listed synonyms (bit$(s; K_1) = 1$).

$\delta_{\text{noun}} = 0.18$, $\delta_{\text{verb}} = 0.34$ (higher because verb synsets are coarser-grained), $\delta_{\text{adj}} = 0$ (adjective similarity too noisy to threshold).

4. Experimental Setup

4.1. Dataset

The training corpus consists of 300 EU Parliament speeches in English (mean 383 words, range 96–690 words, mean 39.8 unique scoreable types per text).

4.2. Evaluation Protocol

We evaluate on the TIRA platform [1]. The platform runs our watermarking container, applies an automated obfuscation attack (sentence shuffling for the spot-check; a stronger attack for the official test), then passes the result to our detection container. TWF is computed as the per-text-pair average of $\max(\text{BLEU}_i, \text{BERTScore}_i) \times \mathbb{1}[\text{correct detection}_i]$.

4.3. Systems Submitted

We submitted four Docker-based systems to TIRA (TIRA submission identifiers in parentheses):

CPU (`milky-bench`) WordNet synonyms only, single key K_1, $p_0 = 0.500$.

GPU v1 (`matching-setter`) Adds BERT-LS semantic fallback, single key K_1, $p_0 = 0.500$.

GPU v2 (`pleasant-box`) BERT-LS fallback, single key K_1, calibrated $p_0 = 0.513$.

Dual-key (`interior-home`) BERT-LS fallback, dual keys K_1, K_2, calibrated p_0.

All model weights are pre-downloaded at Docker build time for TIRA's network-isolated environment.

4.4. Parameters

Table 1 summarises all hyperparameters.

Table 1
System hyperparameters.

Parameter	Value
Detection threshold Z_τ	1.5
Embedding target τ_{target}	2.5
Restoration target τ_{corr}	1.8
Max types per embedding stage M	12
$p_0^{(K_1)}$ / $p_0^{(K_2)}$	0.513 / 0.508
POS categories	NN, NNS, JJ, VB, VBP
Noun / verb similarity floor	0.18 / 0.34
Max WordNet synset depth (noun/verb/adj)	3 / 2 / 1
BERT fill-mask model	`bert-base-uncased`
Sentence embedding model	`all-mpnet-base-v2`
BERT-LS similarity floor	0.45

Table 2
Local ablation on 300 EU Parliament training texts (approximate token-level fidelity; official sentence-level metrics reported in Table 3).

System	TP	TN	BA	BLEU	BERTScore	TWF
Baseline (KGW+LLM)	0.85	0.85	0.85	0.65	0.82	0.70
Single key, $p_0 = 0.5$	1.000	0.897	0.948	0.965	0.988	0.937
Single key, calibrated p_0	1.000	0.920	0.960	0.964	0.988	0.948
Dual key (final)	**1.000**	**1.000**	**1.000**	0.918	≈0.985	**≈0.985**

Table 3
Official TIRA results for all four systems. *Spot-check*: `spot-check-dataset-20260505-training`; *Official test*: `watermarking-20260511-test`. TIRA submission identifiers: CPU = `milky-bench`; GPU v1 = `matching-setter`; GPU v2 = `pleasant-box`; Dual-key = `interior-home` (see Section 4.3). All values are reported directly by TIRA using the official column names: BA = Balanced Accuracy; BLEU and BERTScore are averaged over text pairs; TP = True Positive fraction; TN = True Negative fraction; TWF = Text Watermarking Fidelity (official ranking metric). TP and TN are expressed as fractions of the full evaluation set (watermarked + original texts combined). Best TWF in each dataset partition shown in bold.

System	Dataset	BA	BLEU	BERTScore	TP	TN	TWF
CPU	Spot-check	0.913	0.989	0.485	0.451	0.049	0.926
GPU v1	Spot-check	0.907	0.989	0.498	0.451	0.049	0.938
GPU v2	Spot-check	0.904	0.988	0.497	0.460	0.040	0.945
Dual-key	Spot-check	**0.995**	0.796	0.976	0.496	0.499	**0.971**
CPU	Official test	0.923	0.990	0.292	0.469	0.031	0.753
GPU v1	Official test	0.904	0.988	0.317	0.468	0.032	**0.775**
GPU v2	Official test	0.903	0.988	0.302	0.477	0.023	0.769
Dual-key	Official test	0.753	0.789	0.975	0.254	0.499	0.734

5. Results

Table 2 reports a local ablation on the 300 training texts that traces the incremental contribution of each design decision. Table 3 reports official TIRA results for all four systems on both the spot-check dataset and the official PAN 2026 test set.

5.1. Ablation: Incremental Design Decisions

Type-level detection and POS expansion. Restricting scoreable types to $\{\text{NN}, \text{JJ}\}$ left 28/300 texts undetectable (TP ≈ 0.907); adding $\{\text{NNS}, \text{VB}, \text{VBP}\}$ raised TP to 1.000 by providing sufficient type diversity across all texts.

Empirical null calibration. Correcting the null fraction from 0.500 to 0.513 shifted the Z-distribution for original texts to be centred near zero, reducing false positives from 31 to 24 (TN: 0.897 $\rightarrow$ 0.920) with no change to TP.

Dual-key AND detection. Adding the second key eliminated all remaining false positives on the training set (TN: 0.920 $\rightarrow$ 1.000, Table 2). The cost is approximately $2\times$ more word substitutions per text (14 $\rightarrow$ 31 on average), which reduces BLEU (0.964 $\rightarrow$ 0.918) but leaves BERTScore (≈ 0.985) essentially unchanged in local evaluation, because synonym substitution preserves semantics. In these local ablations BERTScore exceeds BLEU for all systems, so the local fidelity term is dominated by BERTScore; the TIRA evaluation reveals a more nuanced picture (Table 3 and Section 6.2).

Simulated attack robustness. Local simulations of word-substitution attacks show: TP = 0.967 after 20% substitution and TP = 0.863 after 50% substitution. The embedding target $\tau_{\text{target}} = 2.5$ provides a margin above $Z_\tau = 1.5$: a 30% attack is expected to reduce the post-attack Z-score to $\approx 2.5 \times 0.7 = 1.75$, still above threshold. Position-shuffling attacks cause zero Z-score degradation because $\text{bit}(w; K)$ depends only on word identity, not position.

6. Discussion

6.1. The Precision–Robustness Trade-off in AND Detection

Table 3 reveals a stark inversion between the two evaluation settings. On the spot-check dataset (sentence-shuffling attack), the dual-key system is clearly superior: TWF = 0.971, BA = 0.995. On the official test (stronger attack), the dual-key system ranks last: TWF = 0.734, with TP collapsing from 0.496 to 0.254. The single-key GPU v1 system achieves the best official TWF of 0.775 with consistent BA = 0.904.

The mechanism is transparent. The AND condition (Equation 4) fails if the attack drives *either* $Z^{(K_1)}$ or $Z^{(K_2)}$ below $Z_\tau = 1.5$. Under the mild sentence-shuffling attack, both Z-scores are fully preserved: position independence means no signal is lost, and the AND condition holds easily. Under the stronger official attack, word-level substitutions remove embedded green types; even with a safety margin of 1.0 above threshold (embedded at 2.5, threshold 1.5), the official attack is evidently aggressive enough to suppress at least one key in roughly half of watermarked texts, causing detection failure. The single-key system, which requires only one signal to survive, is structurally more resilient to this pattern.

Crucially, the dual-key system's TN remains 0.499 on *both* datasets: it never false-detects. This near-zero false-positive behaviour is maintained at the cost of missed detections once both signals are not simultaneously preserved.

6.2. Fidelity–Robustness Interaction

Dual-key embedding requires $\approx 2\times$ more substitutions than single-key embedding. On the TIRA official test, this reduces dual-key BLEU to 0.789 compared with ≈ 0.989 for single-key systems, while dual-key BERTScore rises to 0.975 compared with 0.292–0.317 for single-key. The dominance of each metric is therefore *system-dependent*: for single-key systems BLEU is the binding fidelity measure (BLEU $\gg$ BERTScore), while for the dual-key system BERTScore is the binding measure (BERTScore $>$ BLEU). In both cases, however, $\max(\text{BLEU}, \text{BERTScore})$ lies in the narrow range 0.975–0.990 across all four systems (see BLEU and BERTScore columns in Table 3), so the fidelity component of TWF is

approximately equal and the differences in final TWF are almost entirely attributable to differences in BA. Further reducing surface-level changes (e.g. preferring rarer synonyms) is therefore unlikely to improve TWF meaningfully; robustness of the detection criterion is the binding constraint.

6.3. Implications for Dual-Key Watermarking Design

The precision–robustness trade-off observed here has a straightforward structural explanation that generalises beyond this specific system. In any AND-condition multi-key scheme, the probability of successful detection is $\prod_i P(\text{key } i \text{ survives attack})$. If the attack independently degrades each key's signal by the same fraction, adding more keys strictly reduces TP. An OR-condition scheme ($Z_1 \geq Z_\tau$ *or* $Z_2 \geq Z_\tau$) restores robustness but at the cost of a higher FP ($P(\text{FP}) \approx 1 - (1 - 0.067)^2 \approx 0.129$ at $Z_\tau = 1.5$). A hybrid scheme — using AND at a lower threshold on a per-text basis, or switching to OR when one key yields a very high Z-score — could recover detection without sacrificing all of the precision gain.

6.4. Limitations

Two limitations remain. First, the AND condition is brittle under the official attack: the dual-key system misses roughly half the watermarked texts after obfuscation, and increasing the embedding target to 3.0 or 3.5 might provide enough margin to survive. Second, all systems rely on WordNet synonym availability; BERT-LS reduces but does not eliminate the gap for words with sparse synonym sets.

7. Conclusion

We presented a post-hoc synonym substitution watermarking system for PAN@CLEF 2026, with two design choices — type-level Z-test detection and dual-key AND criterion — that form the basis of the main empirical contribution.

On the spot-check dataset, the dual-key system achieves the highest performance among all submissions (TWF = 0.971, BA = 0.995, TP = 0.496, TN = 0.499), confirming that the AND criterion almost perfectly suppresses false positives when both signals survive intact. On the official test set, the single-key system achieves the best TWF of 0.775 (BA = 0.904), while the dual-key system drops to TWF = 0.734 (BA = 0.753, TP = 0.254, TN = 0.499).

The key finding is an empirically verified precision–robustness trade-off: the AND condition dramatically reduces FP but introduces brittleness when the attack can suppress either signal independently. This trade-off has a structural explanation applicable to any multi-key AND watermarking scheme, and suggests that future work should explore hybrid AND/OR detection or higher embedding targets to balance TP and FP under strong obfuscation attacks.

Code is publicly available.[1]

Acknowledgments

We thank the PAN 2026 organizers for providing the evaluation infrastructure via TIRA [1] and the benchmark datasets.

Declaration on Generative AI

During the preparation of this work, the author used Claude (Anthropic) in order to: assist with code development and writing assistance. After using this tool, the author reviewed and edited the content as needed and takes full responsibility for the publication's content.

[1] https://github.com/Momen-Ibrahim-Fawzy/posthoc-dualkey-watermarking--PAN2026

References

[1] 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: 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.

[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] 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.

[4] J. Kirchenbauer, J. Geiping, Y. Wen, J. Katz, I. Miers, T. Goldstein, A Watermark for Large Language Models, in: Proceedings of the 40th International Conference on Machine Learning (ICML 2023), PMLR, 2023, pp. 17061–17084. URL: https://proceedings.mlr.press/v202/kirchenbauer23a.html.

[5] S. J. Chang, A. F. Cooper, H. Alvari, J. Zou, P. Liang, T. Hashimoto, PostMark: A Robust Blackbox Watermark for Large Language Models, in: Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP 2024), Association for Computational Linguistics, 2024, pp. 8711–8729. doi:10.18653/v1/2024.emnlp-main.506.

[6] X. Zhao, P. Ananth, L. Li, Y.-X. Wang, Provable Robust Watermarking for AI-Generated Text, in: Proceedings of the 12th International Conference on Learning Representations (ICLR 2024), 2024. URL: https://openreview.net/forum?id=SsmT8aO45L.

[7] M. Topkara, U. Topkara, M. J. Atallah, The Hiding Virtues of Ambiguity: Quantifiably Resilient Watermarking of Natural Language Text through Synonym Substitutions, in: Proceedings of the 8th ACM Workshop on Multimedia and Security, ACM, 2006, pp. 164–174. doi:10.1145/1161366.1161397.

[8] S. Bird, E. Klein, E. Loper, Natural Language Processing with Python, O'Reilly Media, 2009. URL: https://www.nltk.org/book/.

[9] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, in: Proceedings of NAACL-HLT 2019, Association for Computational Linguistics, 2019, pp. 4171–4186. doi:10.18653/v1/N19-1423.

[10] N. Reimers, I. Gurevych, Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks, in: Proceedings of EMNLP-IJCNLP 2019, Association for Computational Linguistics, 2019, pp. 3982–3992. doi:10.18653/v1/D19-1410.