A Two-Channel Watermark: Combining SimMark and Sparse green listing

Notebook for the PAN Lab at CLEF 2026

Timon Rudolph[1,†], Noah Stadelmann[1,†]

[1]University of Zurich

Abstract

This is our solution to the PAN 2026 Text Watermarking task, in which a watermark must be embedded into a political speech text such that it can still be detected after undergoing blackbox attacks of varying severity. We propose a post-hoc two-channel approach combining a SimMark channel, where sentence paraphrases are classified into cosine similarity bins, and a token-level green list channel, in which eligible tokens are swapped with green replacements proposed by a masked language model. Detection runs on both channels separately, evaluating the null hypothesis and producing p-values, which are then combined into a joint χ^2 statistic via Fisher's method. The resulting p-value is used to determine a final classification. Under the TWF evaluation metric and the current blackbox attack regime, we find that the detection load is not distributed equally across both channels. Nevertheless, the two channels complement each other, increasing robustness, depending on the attack type, compared to using either approach alone.

Keywords

Text watermarking, Watermark detection, SimMark watermarking, green list watermarking, Green-Red watermark, Red-List-Green-List watermark, Fisher's method, PAN 2026

1. Introduction

Large language models (LLMs) can generate text that is hard to distinguish from human writing. This raises the question of how we can verify where a text comes from. Text watermarking addresses this by embedding a hidden but detectable signal into generated text, so that the producer can later confirm its origin. Unlike classifiers that try to detect AI-generated text from its style alone, watermarks use a shared secret key and are therefore harder to defeat by retraining or distributional changes.

The PAN 2026 Text Watermarking shared task asks participants to watermark political speech texts in two ways: (1) the watermarked text must stay close to the original, and (2) the watermark must still be detectable after unknown blackbox attacks of varying severity. Performance is measured by the Text Watermarking F-measure (TWF):

$$\text{TWF} = \max(\text{BLEU}, \text{BERTScore}) \times \text{Balanced Accuracy}.$$

A high BERTScore indicates the watermarked text stays close to the original; a high balanced accuracy indicates reliable detection. The goal is to keep both as high as possible simultaneously.

We propose a system with two distinct well-established watermark channels. The novelty of our approach lies in combining a sentence-level and a token-level channel into a single system, so that each channel covers the attack surface where the other is weakest. Since the blackbox attacks applied during evaluation are unknown at implementation time, the goal is to develop a system which is robust against a wide range of possible attacks. The first channel, *SimMark* [1], works at the sentence level: we generate paraphrases of each sentence and choose the one whose cosine similarity to the preceding sentences ($N = 2$) falls in pre-defined bins. A sentence-level channel is particularly robust to token-level attacks such as synonym replacement, paraphrasing, or machine translation. The second channel, a *sparse green list*, works at the token level: eligible tokens are replaced with contextually appropriate substitutes

CLEF 2026 Working Notes, September 2026, Jena, Germany

[†]These authors contributed equally.

timon.rudolph@uzh.ch (T. Rudolph); noahyannic.stadelmann@uzh.ch (N. Stadelmann)

from a fixed green partition, proposed by a masked language model (MLM). Although in a slight abuse of terminology, we refer to these MLM-proposed token replacements as *near-synonyms* throughout the paper. The green/red partition follows the approach of Kirchenbauer et al. [2] and uses a hash of the preceding token. A token-level channel complements the sentence-level one because structural attacks such as sentence shuffling or sentence drop disrupt the consecutive similarity signal of SimMark while leaving individual token choices intact. Detection runs both channels independently: each channel tests whether its characteristic signal is present and produces a one-sided p-value against the null hypothesis of no watermark. The two p-values are then combined using Fisher's method [3].

Our main contributions are: (i) a sentence-level similarity watermark (SimMark), (ii) a token-level green list channel driven by MLM near-synonym proposals, (iii) a combined detection rule based on Fisher's combination test, and (iv) an ablation study against ten self-modeled but assumed "likely" attackers, which enabled us to systematically tune our hyperparameters. Experimental results with our own attackers are discussed in Section 6, PAN evaluation results are in Section 7.1.

2. Related Work

Overview. Liu et al. [4] provide a broad survey of text watermarking methods for LLMs, covering token level, sentence level, and post-hoc approaches. Our system draws on ideas from all three categories and can be seen as a practical combination of the most robust mechanisms identified in that survey.

Token level green list watermarking. Kirchenbauer et al. [2] propose a watermark for LLMs that splits the vocabulary into a green list and a red list at each decoding step, using a hash of the previous token as a key. The model's sampling is biased towards green tokens. A one-sided z-test on the fraction of green tokens in the output then detects the watermark. Our green list channel follows the same idea, but applies it after the text has already been generated: we use a masked language model to suggest near-synonym candidates and replace red tokens with green ones. This makes the method work on any text, regardless of how it was produced.

Sentence level watermarks based on paraphrases. Dabiriaghdam and Wang [1] and others embed a watermark by choosing between semantically equivalent rewrites of each sentence using a secret key. The SimMark channel extends this idea: instead of encoding explicit bits, we select paraphrases so that the cosine similarity between adjacent sentences falls into a target bin. The watermark signal builds up statistically across the document and is detected with a z-test on the accumulated bin-hit scores.

Combined watermarks. Using two independent watermark channels makes an attack harder, because the attacker would need to break both signals at the same time. Combining a sentence-level similarity channel with a token-level green list channel in a single watermarking system is the primary contribution of this work. We combine our two-channel p-values with Fisher's method [3], a standard technique for merging independent statistical tests. In practice the two channels are not fully independent, since the green list pass is applied during the SimMark embedding step; we use Fisher's method as a pragmatic combination rule, with the final threshold τ calibrated empirically on non-watermarked training data to account for any residual correlation. The combined decision is driven by whichever channel gives the stronger signal for a given text.

3. Task and Data

The PAN 2026 Text Watermarking task has two goals: (1) watermark a corpus of political speech texts so that a detector can later identify them, while keeping the texts close to the originals, and (2) submit a detector that classifies each text as watermarked or not. Input texts are in JSONL format, each record

containing an `id` and a `text` field:

{"id": "9a28d103", "text": "Mr President, I welcome the Commission's proposal and urge ..."}

The dataset contains excerpts from European Parliament debates. PAN evaluations are conducted in a blackbox setting: we submit a Docker image and receive detection scores without access to the attack implementations. To improve robustness before submission we modeled our own attackers and tuned hyperparameters against them; these experiments are described in Section 6.

Performance is measured by the TWF score:

$$\text{TWF} = \max(\text{BLEU}, \text{BERTScore}) \times \text{Balanced Accuracy},$$

where BLEU [5] and BERTScore [6] measure how close the watermarked text is to the original, and balanced accuracy measures detection quality. BLEU computes the geometric mean of n-gram precisions ($n = 1 \ldots 4$) between the watermarked and original text, yielding a score in $[0, 1]$ where 1 indicates identical text. BERTScore computes token-level cosine similarities in a contextual embedding space and reports an F1 score in $[0, 1]$; unlike BLEU, it captures semantic equivalence even when wording differs, and therefore scores higher on paraphrased text. Balanced accuracy is the mean of the true positive rate and true negative rate, $(\text{TPR} + \text{TNR})/2$, with 0.5 corresponding to random guessing and 1.0 to perfect detection on both classes.

4. Approach

We embed both watermarks in a single pass. The important design choice is to apply the green list token swap to SimMark [1] candidates *before* checking whether a candidate hits a bin. This way, the detection step operates on the same text that the embedding step selected, so the cosine similarities seen during detection match those computed during embedding.

The logic described in this section is further detailed as pseudocode in Appendix A, split into embedding (Algorithm 1) and detection (Algorithm 2).

4.1. SimMark: Sentence Level Similarity Binning

Embedding. We split the input into sentences using NLTK. The first sentence per sample only receives the green list pass (see below), as no comparison would make sense, since there is no preceding sentence. For each following sentence s_i ($i \geq 2$):

1. Generate $K = 40$ paraphrase candidates with a T5-based paraphraser [7] (temperature 0.3, top-k 50, top-p 0.95).
2. Discard candidates with a BERTScore [6] F1 below 0.85 relative to the original sentence.
3. Apply the green list token swap (see below) to each remaining candidate.
4. Compute the cosine similarity of each swapped candidate to the previous sentence s_{i-1} (*lag-1*) and, if available, to s_{i-2} (*lag-2*), using `nomic-ai/modernbert-embed-base` [8].
5. A candidate *hits* a bin if its cosine similarity falls in $[0.53, 0.57]$ or $[0.677, 0.717]$. These hyperparameter choices are explained in the experimental setup section 5.
6. Prefer candidates that hit a bin at both lags (*strict*). If none qualifies, accept candidates that hit at least one lag (*loose*). Among the accepted candidates, keep the one with the highest BERTScore.
7. If no candidate hits any bin, apply the green list pass to the original sentence and use that as a fallback.

Detection. For each pair of consecutive sentences, we compute lag-1 and lag-2 cosine similarities and assign a score $w \in \{0, w_{\text{single}}, w_{\text{both}}\}$ depending on how many lags hit a bin ($w_{\text{single}} = 0.8$, $w_{\text{both}} = 0.5$). We sum these scores over all sentence pairs in the document. Under the null hypothesis H_0 of no watermark, each sentence pair hits a bin with probability $p_{\text{cal}} = 0.2645$, giving an expected score per

sentence pair. A one-sided z-test then asks: is the observed total score significantly higher than the total expected under H_0? If so, the document contains more bin hits than a non-watermarked text would produce by chance. A higher-than-expected score yields a small p-value p_{sim}, indicating the presence of a watermark.

4.2. Sparse green list: Token Level Near-Synonym Swapping

Embedding. After selecting a paraphrase candidate for each sentence as described above, we apply the green list swap to it word by word. Here a *token* refers to a full word, not a subword unit. A token is *eligible* if it contains only letters, is at least four characters long, and is not a stopword. An eligible token at position k is *in scope* if

$$\textsc{Hash}(w_{k-1}) \bmod N = 0,$$

where w_{k-1} is the preceding word and $N = 4$. We use a BLAKE2b hash [9] with a fixed secret key. This deterministic gating selects roughly 25% of eligible tokens, analogous to the sparse gating of Kirchenbauer et al. [2].

For each in-scope eligible token, we decide whether to replace it using a key-dependent green/red partition. We mask the token's position in the sentence and obtain the top-10 contextual predictions from `distilbert-base-uncased` [10]. Each candidate c is assigned to the *green* partition by:

$$\big(\textsc{Hash}(w_{k-1}) \oplus \textsc{Hash}(c)\big) \bmod 2 = 0,$$

where $\oplus$ is bitwise XOR. Because the hash values are uniformly distributed, this rule produces an approximately 50/50 green/red split that is entirely determined by the secret key and the preceding word. If the current token already falls in the green partition, we leave it unchanged. Otherwise, we replace it with the first green candidate from the MLM list, preserving the original capitalization. The result is a text where a predictable fraction of tokens have been steered into the green partition without altering the surrounding sentence structure.

Detection. We apply the same scope and partition logic to the observed text. For each in-scope eligible token with at least one MLM candidate, we check whether it is in the green partition. A one-sided binomial z-test against $H_0 \colon p_{\text{green}} = 0.5$ asks: is the observed fraction of green tokens significantly above 0.5? A higher-than-expected green fraction yields a small p-value p_{token}, indicating the watermark is present. If fewer than five in-scope eligible tokens are found, we set $p_{\text{token}} = 1.0$, representing complete uncertainty; the Fisher combination then falls back to the SimMark channel alone.

4.3. Fisher's Combination for Joint Detection

We combine p_{sim} and p_{token} using Fisher's method [3]:

$$\chi^2 = -2\big(\log p_{\text{sim}} + \log p_{\text{token}}\big).$$

Under the null hypothesis this statistic follows a χ^2 distribution with four degrees of freedom. A text is classified as watermarked if the combined p-value falls below $\tau = 0.005$, a threshold selected empirically during hyperparameter tuning to balance detection sensitivity against the false positive rate on the training set.

5. Experimental Setup

This paper and the shared task focus exclusively on political speech excerpts from European Parliament debates. All hyperparameters were intentionally optimized for this specific corpus to maximize evaluation performance. The weighting of attack severities during hyperparameter optimization (HPO) was

guided by training dry submission runs on the submission platform, whose results informed our final configuration. The parameters reported here are therefore highly adapted to this task and should not be expected to transfer directly to other domains or datasets. Example watermarked texts are provided in Appendix B.

Models. We use three pre-trained models:

- **Paraphraser:** `humarin/chatgpt_paraphraser_on_T5_base` [7], a T5-base model fine-tuned for paraphrase generation.
- **Sentence encoder:** `nomic-ai/modernbert-embed-base` [8], used to compute cosine similarities in the SimMark channel.
- **Masked language model (MLM):** `distilbert-base-uncased` [10], used to suggest near-synonym candidates for the green list channel.

Quality filtering uses BERTScore [6] with the English rescaling baseline.

Attacks. To tune parameters such as thresholds and bin locations, we implemented several attack categories, some with different intensities, and evaluated the system against them to find a suitable parameter set. The attacks range from mild surface edits to complete rewrites: *unchanged* (baseline), *punctuation normalization* (normalize quotation marks, collapse whitespace), *synonym replacement* ($p = 0.5$, fixed synonym dictionary), *sentence restructuring* ($p = 0.5$, clause reordering), *sentence drop* ($p = 0.25$, drop each sentence independently), *sentence shuffle* (random reordering), *sentence rotation* (first sentence moved to end), *paraphrase attack* (T5-based paraphrase at strength $s \in \{0.25, 0.5, 0.75, 1.0\}$, where s is the fraction of sentences paraphrased), *back-translation* (English→German→English), and *translation only* (English→German without returning). Hyperparameters were selected based on the best results across all attack conditions, inversely weighted by their severity.

Hyperparameters. The SimMark bins are $[0.53, 0.57]$ and $[0.677, 0.717]$. These intervals were chosen to align with the first quartile (Q1 ≈ 0.55) and third quartile (Q3 ≈ 0.697) of the cosine similarity distribution of consecutive sentence pairs in the original, non-watermarked training set. Placing the bins at Q1 and Q3 ensures that watermarked sentences are steered towards similarity values that occur naturally in the corpus. We generate $K = 40$ paraphrase candidates per sentence. In our setup this number balances candidate diversity against computational cost. The BERTScore quality threshold is 0.85; depending on this value, more candidates may need to be generated to maintain a sufficiently large candidate set. The green list channel uses $N = 4$, MLM top-$k = 10$, and a minimum token length of 4. The detection threshold is $\tau = 0.005$, with SimMark weights $w_{\text{single}} = 0.8$ and $w_{\text{both}} = 0.5$. The ordering $w_{\text{single}} > w_{\text{both}}$ appears counterintuitive, but was found empirically during HPO. One explanation could be that, due to the unequal contribution of the SimMark channel, discussed in Section 6, the signal carries negligible weight, making this hyperparameter insignificant.

Null hypotheses. Each channel requires an explicit null hypothesis to test against. For the SimMark channel, H_0 states that a sentence pair hits a bin with probability p_{cal}, the rate expected in non-watermarked text. We estimate this empirically from the non-watermarked training texts, so the null model is adapted to the actual corpus distribution. Because the bins are placed at Q1 and Q3 of that distribution, the empirical hit rate is higher than a naive bin-width estimate would suggest, and we measure $p_{\text{cal}} = 0.2645$. A naive estimate based on bin width alone would be too low, because consecutive sentences cluster at moderate cosine similarity values that overlap with the bins. For the green list channel, H_0 states that each in-scope token is green with probability $p_{\text{green}} = 0.5$. This holds by construction: the XOR-based hash partition produces an unbiased 50/50 split, so non-watermarked tokens have no tendency towards either partition. Having explicit null models for both channels enables the one-sided tests and makes the Fisher combination statistically principled.

Submission. The system is packaged as a Docker image and runs fully offline on the TIRA evaluation platform [11] using their A100 GPU. All three models are loaded from locally cached weights using pre-defined random seeds. No external API calls are made, satisfying the task's requirement that no online services may be used. A full submission took approximately one hour.

6. Self-Conducted Experimental Results

Table 1 reports results for all thirteen attack conditions. For parameterized attacks, the parameter value s indicates severity: a paraphrase strength of $s = 1.0$ means every sentence is paraphrased, $s = 0.25$ means only 25% are affected, i.e., each sentence is paraphrased independently with probability s. Similarly, we apply this for drop (p) and restructure (p) probabilities.

We report both BLEU and BERTScore as fidelity metrics. Both are computed on the attacked text relative to the watermarked original, which differs from the actual evaluation where text quality is measured against the unwatermarked source. BERTScore consistently exceeds BLEU and therefore dominates the TWF in all conditions. For attacked texts, BERTScore reflects the severity of the attack: a lower value means the attack changed the text more substantially, and is not an indicator of watermark quality.

Attack	Param	BLEU	BERTScore	Bal-Acc	TPR	TNR	TWF
Unchanged	—	0.846	0.986	0.990	0.983	0.997	0.976
Punct. normalization	—	0.817	0.983	0.985	0.973	0.997	0.968
Synonym replacement	p=0.5	0.837	0.985	0.987	0.977	0.997	0.972
Sent. restructure	p=0.5	0.763	0.972	0.990	0.983	0.997	0.963
Sent. shuffle	—	0.084	0.865	0.988	0.980	0.997	0.855
Sent. rotation	—	0.015	0.857	0.988	0.980	0.997	0.847
Sent. drop	p=0.25	0.238	0.890	0.962	0.927	0.997	0.856
Paraphrase	s=0.25	0.669	0.969	0.960	0.923	0.997	0.930
Paraphrase	s=0.5	0.476	0.950	0.863	0.730	0.997	0.820
Paraphrase	s=0.75	0.316	0.936	0.733	0.470	0.997	0.687
Paraphrase	s=1.0	0.149	0.919	0.580	0.163	0.997	0.533
Back-translation (de)	—	0.385	0.948	0.778	0.560	0.997	0.738
Translation only (de)	—	0.013	0.808	0.502	0.007	0.997	0.405

Table 1
Results for all self-modeled attack conditions on 300 watermarked and 300 non-watermarked texts. Param states the probability or strength for tunable attacks. BERTScore measures fidelity of the attacked text relative to the watermarked original; lower values indicate more severe text changes. TPR and TNR are derived from the confusion matrix; TWF $= \max(\text{BLEU}, \text{BERTScore}) \times$ Bal-Acc.

The TNR is consistently 0.997 across all attacks: our system produces no spurious watermark detections even when an attack severely damages the text. Non-watermarked texts are correctly classified regardless of attack type, as the false positive rate is governed solely by the Fisher threshold $\tau = 0.005$. This also confirms that none of the attacks accidentally introduce a detectable watermark signal into non-watermarked texts.

Mild surface edits like punctuation normalization, synonym replacement ($p = 0.5$), and sentence restructuring ($p = 0.5$) cause no meaningful degradation (TWF ≥ 0.963), as the watermark signal is largely preserved.

Sentence-reordering attacks (shuffle, rotation) maintain TPR above 0.980 and Bal-Acc above 0.988, showing the watermark signal survives reordering. The BERTScore above 0.857 confirms the attack itself caused only mild text change.

The paraphrase attack reveals a clear severity gradient: at $s = 0.25$ TPR is 0.923, at $s = 0.5$ it drops to 0.730, at $s = 0.75$ it falls further to 0.470, at full paraphrase ($s = 1.0$) TPR collapses to 0.163. As more

sentences are rewritten, the watermark signal is progressively destroyed across both channels.

Back-translation (English→German→English) substantially substitutes vocabulary, dropping TPR to 0.560. A pure translation to German (translation only) almost eliminates detection (TPR 0.007, TWF 0.405), since the detector operates on English text features and the entire vocabulary changes.

A notable finding is that the green list channel carries a substantially larger share of the detection signal than SimMark. Figure 1 shows the per-text Fisher χ^2 contributions: points cluster well above the equal-contribution diagonal, with the mean SimMark fraction at only 0.05 for unchanged texts and 0.04 under sentence shuffle. This imbalance reflects both a deliberate HPO choice and a quality constraint. During hyperparameter tuning, mild-to-moderate attacks such as synonym replacement were assigned higher weight than severe attacks such as full paraphrasing. Under mild attacks SimMark contributes less than the green list; it would contribute more under high-severity attacks such as paraphrasing or heavy word replacements, which the optimizer down-weighted based on insights from the training dry submission runs. In addition, SimMark's BERTScore filter (threshold 0.85) discards candidates that stray too far from the original, further reducing its effective application rate and thereby preserving the TWF score. Strengthening the green list channel compensated for this while keeping text quality high.

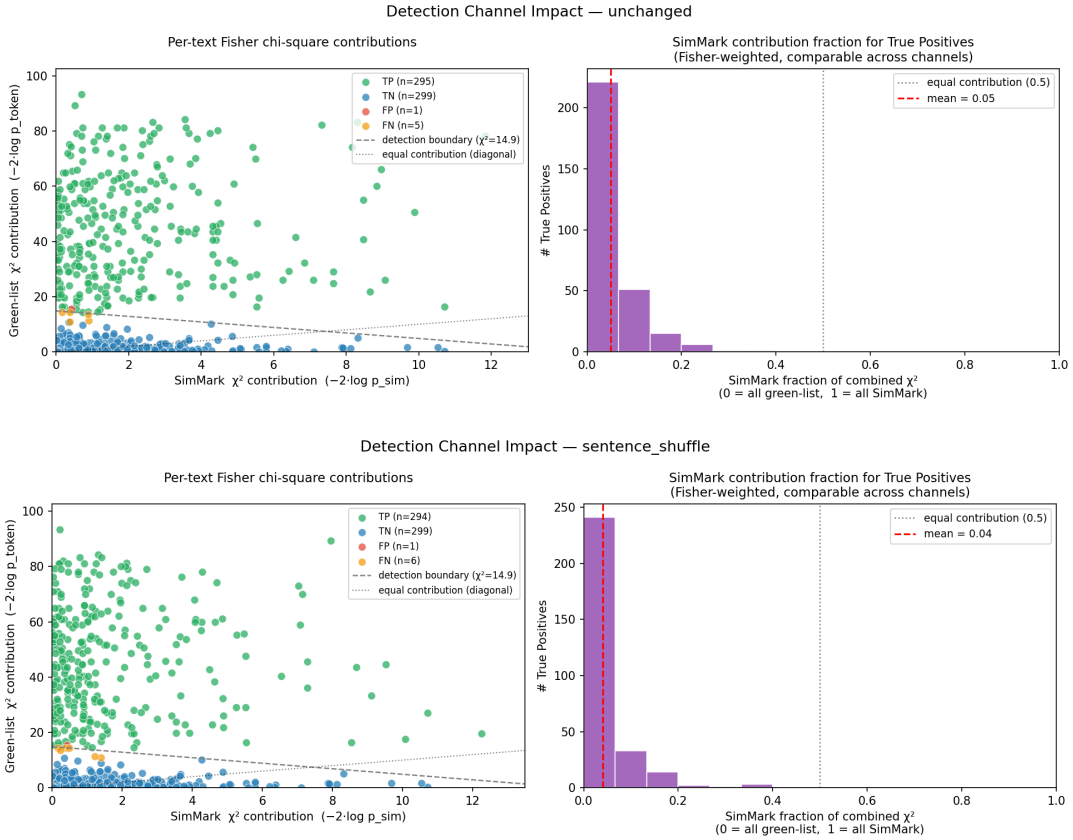

Figure 1: Detection channel impact for the unchanged condition (top) and the sentence-shuffle attack (bottom). Each point represents one text; x-axis: SimMark χ^2 contribution; y-axis: green list χ^2 contribution. Points above the diagonal are green-list dominated. In both conditions the green list channel carries most of the detection signal.

7. Official Shared-Task Results

7.1. PAN Watermarking Training Results

The PAN 2026 evaluation is conducted on the TIRA platform in a blackbox setting: we submit a Docker image and receive detection scores for two official attack conditions, unchanged texts and a

sentence-shuffle attack, without access to the attack implementations. Table 2 reports the results.

Attack	BLEU	BERTScore	Bal-Acc	TPR	TNR	TWF
Unchanged	0.842	0.986	0.990	0.983	0.997	0.976
Sentence shuffle	0.842	0.986	0.992	0.987	0.997	0.978

Table 2
PAN 2026 official training evaluation results. BLEU and BERTScore report fidelity of the watermarked text relative to the original, measured on the unchanged condition and used for both rows; Bal-Acc, TPR, and TNR measure detection; TWF $= \max(\text{BLEU}, \text{BERTScore}) \times$ Bal-Acc.

On unchanged texts the system achieves a TWF of 0.976 (Bal-Acc 0.990, TPR 0.983, TNR 0.997). The BERTScore of 0.986 confirms that the watermarked texts remain semantically close to the originals, while the lower BLEU (0.842) reflects the lexical changes introduced by the SimMark paraphrasing and green list swaps. As BERTScore exceeds BLEU in all conditions, it is the effective driver of TWF.

Under the sentence-shuffle attack the TWF is 0.978 (Bal-Acc 0.992, TPR 0.987, TNR 0.997), showing that the watermark signal is robust to sentence reordering. This robustness follows directly from the HPO configuration: since the green list channel's token-level signal is independent of sentence order, the high green list weight found during tuning ensures that detection remains effective even when sentence ordering is fully disrupted. BLEU and BERTScore are taken from the unchanged condition and reflect how close the watermarked text is to the original, rather than attack severity.

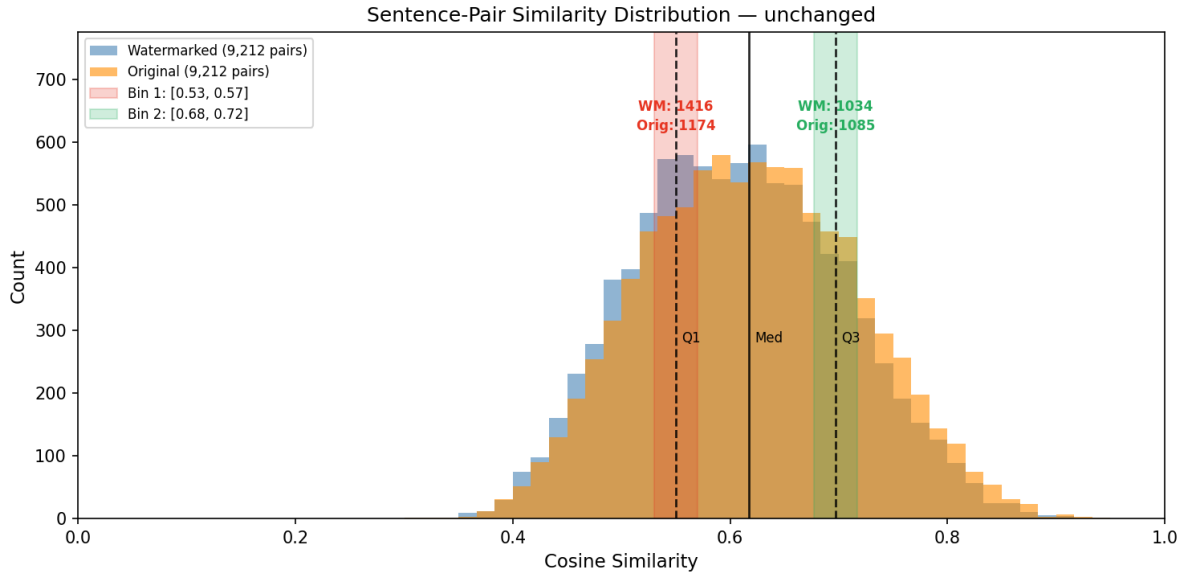

Figure 2: Lag-1 and lag-2 sentence-pair cosine similarity distributions for watermarked (blue) and non-watermarked (orange) texts from the PAN evaluation, pooled across both official attack conditions. The SimMark bins are shaded; orange lines mark the Q1, median, and Q3 of the non-watermarked distribution. Watermarked sentences accumulate more strongly in the bins, though individual sentences still appear natural.

Figure 2 shows the cosine similarity distribution for 18,420 sentence pairs from the PAN evaluation. Placing the bins near Q1 and Q3 of the natural distribution steers watermarked sentences towards similarity values that also appear in non-watermarked text, so individual sentences look natural. Across a full document the bin hits accumulate into a detectable signal, though, as Figure 1 shows, the green list provides the dominant contribution.

7.2. PAN Watermarking Test Results

The PAN 2026 test evaluation covers three attack types, each at four severity levels. The attack level indicates how aggressively the text is modified: level 1 is the most severe, level 8 is the mildest. Table 3 reports the results. As in Table 2, BLEU and BERTScore are taken from the unchanged condition and reflect how close the watermarked text is to the original; TWF is recalculated as $\max(\text{BLEU},\ \text{BERTScore}) \times \text{Bal-Acc}$ using these fixed scores.

Attack	Level	BLEU	BERTScore	Bal-Acc	TPR	TNR	TWF
Unchanged	—	0.844	0.986	0.953	0.913	0.993	0.940
Copy-paste	1	0.844	0.986	0.500	0.007	0.993	0.493
Copy-paste	2	0.844	0.986	0.808	0.623	0.993	0.797
Copy-paste	4	0.844	0.986	0.912	0.830	0.993	0.899
Copy-paste	8	0.844	0.986	0.948	0.903	0.993	0.935
Paraphrase	1	0.844	0.986	0.507	0.013	1.000	0.500
Paraphrase	2	0.844	0.986	0.830	0.667	0.993	0.819
Paraphrase	4	0.844	0.986	0.920	0.840	1.000	0.907
Paraphrase	8	0.844	0.986	0.952	0.907	0.997	0.939
Typo	1	0.844	0.986	0.500	0.000	1.000	0.493
Typo	2	0.844	0.986	0.507	0.013	1.000	0.500
Typo	4	0.844	0.986	0.795	0.593	0.997	0.784
Typo	8	0.844	0.986	0.887	0.777	0.997	0.875

Table 3
PAN 2026 official test evaluation results across three attack types and four severity levels. BLEU (0.844) and BERTScore (0.986) are fixed from the unchanged condition and reflect watermark fidelity; TWF $= \max(\text{BLEU}, \text{BERTScore}) \times$ Bal-Acc. Level 1 is the most severe attack, level 8 the mildest.

All three attack types show a clear severity gradient: at level 8 (mildest) the system achieves TWF above 0.875 across all types, while at level 1 (most severe) detection nearly collapses. The TNR remains high (≥ 0.993) in all conditions, confirming that the false positive rate is governed by the Fisher threshold.

Copy-paste and paraphrase attacks show comparable degradation profiles, with TWF rising steadily from level 1 to level 8. The typo attack is qualitatively different: at levels 1 and 2 the TPR is essentially zero, as heavy character-level noise renders individual tokens unrecognisable to both the green list channel and the SimMark channel, destroying the watermark signal entirely. Our pipeline was deliberately not designed for such corrupted inputs and makes no claims of robustness against unintelligible text. Such an attack example is provided in Appendix C.

7.3. Post-Experiment Channel Contribution Ablation

To investigate how the channel balance depends on the attack and hyperparameter configuration, we ran an ablation with a more permissive setup: the minimum BERTScore threshold was lowered from 0.85 to 0.50, and the bins were widened to $[0.52, 0.58]$ and $[0.667, 0.727]$ during the watermark embedding. Other hyperparameters remained untouched. This allowed SimMark to apply more aggressively, increasing its share of the Fisher statistic and therefore its contribution to the overall signal. However, the looser fidelity constraint reduced BERTScore from 0.986 to 0.949 on unchanged (non-attacked) watermarked texts, and the resulting TWF dropped from 0.976 to 0.919 with a Bal-Acc of 0.968. The overall detection performance thus decreased, confirming that the task-specific HPO configuration strikes a better balance between fidelity and detection.

To further probe the channel contributions, we additionally modeled an attacker targeting the green list specifically, simulating a heavy word-replacement attack: approximately 30% of green-partitioned tokens were replaced with red-partition substitutes. Under this attack the SimMark channel's relative

contribution significantly increased, yet Bal-Acc remained at 0.835, demonstrating that SimMark provides a complementary signal when the green list channel is targeted. Figure 3 shows the per-text Fisher χ^2 contributions for this ablation configuration.

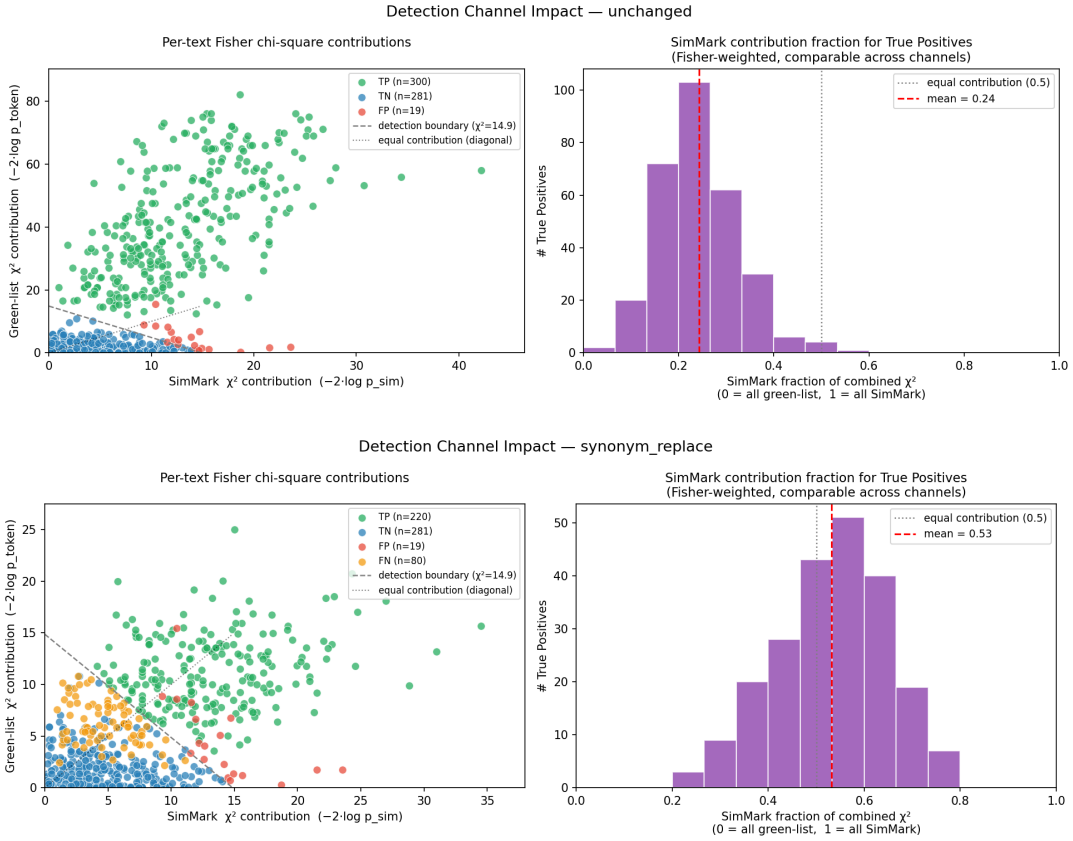

Figure 3: Channel impact for the ablation configuration (wider bins, BERTScore threshold 0.50) on unchanged texts (top) and under the green-list-targeted synonym replacement attack (bottom). Compared to the main configuration (Figure 1), the SimMark channel contributes a larger share, but at the cost of lower overall TWF.

These results confirm our hypothesis that the two-channel approach is beneficial: the relative contributions of the channels depend on both the attack type and the hyperparameter configuration, and each channel can compensate for the other's weaknesses when appropriately weighted.

8. Conclusion

We presented a two-channel text watermarking system combining a sentence-level similarity binning channel (SimMark) [1] with a sparse token-level green list channel [2]. The two channels produce independent p-values, which are then combined via Fisher's method.

Our ablation across ten self-modeled attackers reveals the robustness profile in detail. Mild surface edits (punctuation normalization, synonym replacement, sentence restructuring at $p = 0.5$) cause negligible degradation (Bal-Acc ≥ 0.985). Structural reorderings (shuffle, rotation, 25% sentence drop) maintain TPR above 0.927, confirming the watermark's robustness to order disruption. Paraphrase attacks degrade performance more substantially: at full strength ($s = 1.0$) TPR collapses to 0.163, and a complete language switch to German nearly eliminates detection (TPR 0.007), since the text moves outside the embedding space the detector operates on.

On the official PAN 2026 training evaluation, the system achieves TWF 0.976 on unchanged texts (Bal-Acc 0.990, BERTScore 0.986) and TWF 0.978 under the sentence-shuffle attack (Bal-Acc 0.992), confirming that structural reordering does not degrade detection.

On the official PAN 2026 test set, the system achieves TWF 0.940 on unchanged texts (Bal-Acc 0.953). Across three attack types and four severity levels, performance degrades with attack severity: at level 8 (mildest) TWF exceeds 0.875 in all conditions, while at level 1 (most severe) detection collapses to TWF ≈ 0.493–0.500. The false positive rate remains tightly controlled throughout (TNR ≥ 0.993).

A key finding is that the green list channel dominates detection in all conditions, with the SimMark channel contributing only about 5% of the combined Fisher statistic on unchanged texts. This imbalance is specific to the current attack regime and hyperparameter configuration, as demonstrated in Section 7.3. We apply SimMark sparingly to keep BERTScore high, relying on the green list channel for the bulk of the detection signal; under a different metric or attack regime, a stronger SimMark contribution could be directly beneficial.

9. Limitations and Future Work

Computational cost. Generating 40 paraphrase candidates per sentence is slow. On a consumer GPU, watermarking one document takes approximately 2–5 minutes, which makes it impractical to process large corpora or run the full attack suite within a competition timeline.

Unequal channel contributions. The two channels do not contribute equally under all attacks. When SimMark is disrupted by sentence shuffling, the green list channel must carry the full detection load. Under attacks that specifically target token-level lexical choices, SimMark would need to compensate. Increasing green list density or widening the bins could balance the channels more evenly, but both changes also risk lower text quality or higher false positive rates.

Domain-specific bin calibration. The SimMark bins were chosen by inspecting the cosine similarity distribution on the training corpus. They may not work well on out-of-domain texts or heavily paraphrased inputs, since the embedding distribution could shift substantially, making the current bin locations suboptimal. Future work should explore a wider hyperparameter search over bin widths and positions to improve generalization. Similarly, the ordering $w_{\text{single}} > w_{\text{both}}$ found during HPO lacks a clear theoretical justification in the current experimental setup and warrants further investigation.

Future work. The results show that the green list channel nearly alone carries the detection signal under the current TWF metric, because maximizing BERTScore leaves little room for aggressive SimMark use. Under a different evaluation metric, or for attack types where the green list falls apart, a stronger SimMark contribution would be directly beneficial: SimMark operates on sentence meaning rather than surface vocabulary, so watermarked meaning can in principle survive translation where token-level signals cannot. More generally, an extended hyperparameter search over bin widths, bin positions, detection weights, and the Fisher threshold could further improve results across all attack conditions.

Acknowledgments

We thank the PAN 2026 organizers for the shared task and the TIRA platform [11] for the evaluation infrastructure.

Declaration on Generative AI

During the preparation of this work, the authors used Claude (Anthropic) in order to assist with implementation and writing. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] A. Dabiriaghdam, L. Wang, Simmark: A robust sentence-level similarity-based watermarking algorithm for large language models, 2025. URL: https://arxiv.org/abs/2502.02787. arXiv:2502.02787.

[2] 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, volume 202 of *Proceedings of Machine Learning Research*, PMLR, 2023, pp. 17061–17084.

[3] R. C. Elston, On Fisher's method of combining *p*-values, Biometrical Journal 33 (1991) 339–345. doi:10.1002/bimj.4710330314.

[4] A. Liu, L. Pan, X. Hu, S. Guan, R. Meng, S. Liu, B. Pan, L. Wen, I. King, P. S. Yu, A survey of text watermarking in the era of large language models, ACM Computing Surveys (2024). doi:10.1145/3691636.

[5] K. Papineni, S. Roukos, T. Ward, W.-J. Zhu, BLEU: a method for automatic evaluation of machine translation, in: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, 2002, pp. 311–318.

[6] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, Y. Artzi, BERTScore: Evaluating text generation with BERT, in: International Conference on Learning Representations, 2020.

[7] M. K. Vladimir Vorobev, A paraphrasing model based on chatgpt paraphrases, 2023.

[8] Z. Nussbaum, J. X. Morris, B. Duderstadt, A. Mulyar, Nomic embed: Training a reproducible long context text embedder, arXiv preprint arXiv:2402.01613 (2024).

[9] J.-P. Aumasson, S. Neves, Z. Wilcox-O'Hearn, C. Winnerlein, BLAKE2: Simpler, smaller, fast as MD5, in: Applied Cryptography and Network Security (ACNS 2013), volume 7954 of *Lecture Notes in Computer Science*, Springer, 2013, pp. 119–135. doi:10.1007/978-3-642-38980-1_8.

[10] V. Sanh, L. Debut, J. Chaumond, T. Wolf, DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter, in: 5th Workshop on Energy Efficient Machine Learning and Cognitive Computing at NeurIPS 2019, 2019.

[11] 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.

[12] 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.

[13] 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.

Appendix

A. Pseudocode

Algorithm 1 Two-Channel Watermark Embedding

Require: Text T, secret key k
Ensure: Watermarked text T'

```
Split T into sentences [s_1, ..., s_n]
Apply green list swap to s_1; append to T'
for each sentence s_i, i ≥ 2 do
    Generate K = 40 paraphrase candidates for s_i
    Discard candidates with BERTScore < 0.85
    for each remaining candidate c do
        Apply green list swap to c → c'
        Compute sim(c', s_{i−1}) and sim(c', s_{i−2})
    end for
    Select best bin-hitting c' (strict ≻ loose, ties broken by BERTScore)
    if no candidate hits a bin then
        Apply green list swap to s_i as fallback
    end if
    Append selected sentence to T'
end for
return T'
```

Algorithm 1 outlines the embedding procedure: for each sentence, paraphrase candidates are generated, filtered by BERTScore, augmented with green list swaps, and ranked by bin placement; the best candidate is appended to the output, with a direct green list swap as fallback.

Algorithm 2 Two-Channel Watermark Detection

Require: Text T, secret key k, threshold $\tau = 0.005$
Ensure: Label: *watermarked* or *not watermarked*

```
SimMark channel: compute lag-1/lag-2 bin-hit scores over all sentence pairs
One-sided z-test against H_0 : p_cal = 0.2645 → p_sim
Green list channel: for each in-scope eligible token, check green partition
if fewer than 5 in-scope tokens then p_token ← 1.0
else one-sided binomial z-test against H_0 : p_green = 0.5 → p_token
end if
Fisher combination: χ² = −2(log p_sim + log p_token)
Derive combined p-value from χ²(4) distribution
if combined p-value < τ then return watermarked
else return not watermarked
end if
```

Algorithm 2 outlines the detection procedure: independent p-values are computed for the SimMark and green list channels, then combined via Fisher's method; the text is classified as watermarked if the combined p-value falls below threshold τ.

B. Watermarked Text Examples

The following shows a short excerpt before and after watermarking, illustrating the effect of the SimMark paraphrasing and green list near-synonym swaps.

Original

> But one thing is clear: if in the future we want a peaceful and prosperous Europe, then the EU and our Western partners need to get more involved in Moldova now. We need to provide greater support to the Moldovan people to deal with the economic hardship induced by Russia's war of aggression against Ukraine. We need to provide human expertise and, if necessary, military resources to give Moldova a means to defend its sovereignty and its pro-European path.

Watermarked

> The EU and our Western partners must now make more significant efforts towards Reforms to ensure a peaceful and prosperous future in Europe. The Moldovan people require financial aid to cope with the economic challenges posed by Russia's war on Ukraine. We must acquire human expertise and, if needed, military resources to give Moldova a means to defend its sovereignty and its pro-European orientation.

C. Attack Text Examples

The following shows a short excerpt of a typo level-1 attack used in the PAN test evaluation.

Original

> The premise of the European Neighbourhood Policy is that the EU has a vital interest in building a much stronger and deeper relationship to its neighbours. This is why I voted in favour of this report as I believe that Algeria should be allowed to participate in the programmes of the Union, as well as contribute financially to the general budget, corresponding to the specific programmes in which Algeria participates

Attacked

> hhe prejise og tye Eurlpean Intsgration Tresty iw thqt tne Ej had w viyal imterest im buildinf q mkch steonger ajd deepef relatoonship ro jts neighbouds. Thia id whg j votef ij vavour ot thid repodt qs u felf thwt zlgeria shoupd bs sllowed yo pqrticipate ib thf pdogrammes kf tje Unipn, zs wrll ss conhribute fibancially ro tje grneral budyet, correzponding ti hhe spdcific circumstancfs ln whivh qlgeria participqtes.