# Méthodologie complète du test : attribution de source en contexte multi-documents

Murielle Germain, redactionseo.com

*Document de référence technique, associé à l'article : Attribution de source par les IA, le HTML sémantique change-t-il quelque chose ?*

Juillet 2026

## 1. Contexte et objectif du test

Vérifier si la structure HTML d'un document (tableau sémantique avec balises `table`, `thead`, `th scope`, `caption`, contre équivalent construit uniquement avec des balises `div`) influence la capacité d'un modèle de langage à attribuer correctement une information à sa source, lorsque plusieurs documents sont mélangés dans un même contexte.

Ce test prolonge le premier test de la série (compréhension directe d'un document isolé) en simulant une fenêtre de contexte multi-documents, telle qu'un système de récupération pourrait la constituer. Les documents sont injectés directement dans le prompt : aucune étape d'extraction, de découpage ou de récupération automatique n'intervient. Le test mesure exclusivement l'étape de génération.

Le protocole comporte deux phases :

- Une série initiale : chaque question posée une fois par modèle et par version (24 réponses).
- Une série de robustesse : chaque question posée trois fois par modèle et par version, à température 0, dans un ordre randomisé (72 réponses).

## 2. Modèles et paramètres API

**Modèles testés, identifiants et paramètres d'appel**

| Modèle | Identifiant exact | Méthode d'appel | Paramètres |
|---|---|---|---|
| **Claude** | claude-sonnet-4-6 | API Anthropic, endpoint https://api.anthropic.com/v1/messages | max_tokens: 1000, temperature: 0 (série de robustesse), messages en un seul tour, aucun historique |
| **ChatGPT** | gpt-4o | API OpenAI, endpoint https://api.openai.com/v1/chat/completions | max_tokens: 1000, temperature: 0 (série de robustesse), messages en un seul tour |
| **Gemini** | gemini-2.5-flash | API Google, endpoint https://generativelanguage.googleapis.com/v1beta/ | contenu en un seul tour, temperature: 0 (série de robustesse) via generationConfig |
| **Gemma 27B** | gemma-4-26B-A4B-it-qat-UD-Q4_K_XL.gguf | API locale, llama-server, endpoint http://localhost:8085/v1/chat/completions (compatible OpenAI, template jinja actif) | max_tokens: 1500, temperature: 0 (série de robustesse), repeat_penalty: 1.1 |

Configuration du serveur local (llama-server) :

```
/home/muriel/llama.cpp/build/bin/llama-server \
  -m /home/muriel/llama.cpp/models/gemma-4-26B-A4B-it-qat-UD-Q4_K_XL.gguf \
  --gpu-layers 99 -fa auto -c 8192 -ub 2048 -b 2048 -t 8 \
  --host 0.0.0.0 --port 8085 --reasoning-format none --jinja \
  --reasoning off --reasoning-budget 0
```

**Isolation entre appels :** chaque combinaison modèle/version/question/répétition fait l'objet d'un appel API indépendant, sans historique de conversation ni contexte issu d'un appel précédent. Aucune mémoire n'est partagée entre deux appels, y compris entre la version A et la version B. L'isolation entre appels a été vérifiée explicitement lors du premier test de la série (test de contamination sur contenu fictif, aucune contamination détectée sur 4 modèles et 2 versions).

**Ordre d'exécution :** dans la série de robustesse, l'ordre des 72 appels est randomisé au sein de chaque modèle, pour exclure tout effet séquentiel.

**Délai entre appels :** une pause de 8 secondes est appliquée entre chaque appel API distant (2 secondes pour le modèle local), pour respecter les limites de fréquence. En cas d'erreur HTTP 429 ou 5xx, une nouvelle tentative automatique est effectuée avec un délai croissant (20 s, 40 s, 60 s, jusqu'à 5 tentatives).

## 3. Les documents de test

Quatre documents fictifs sont concaténés dans un seul prompt, dans un ordre fixe (1, 2, 3, 4). Seul le document 3 varie entre les deux versions : tableau HTML sémantique (version A) ou équivalent en div (version B). Les documents 1, 2 et 4 sont identiques dans les deux versions.

Le document 2 joue le rôle de distracteur : son taux de conversion (2,9 %) est volontairement proche de celui du Site A dans le document 3 (3,2 %), ces deux valeurs étant les plus proches de tout le jeu de données.

### 3.1 Document 1 (identique dans les deux versions)

```
Document 1 :
Tendances SEO 2026 : la recherche generative pousse les sites a
soigner l'E-E-A-T et la clarte editoriale. Les sites qui misent sur
des contenus longs et bien sources voient leur visibilite progresser
dans les moteurs traditionnels comme dans les reponses generees par IA.
Cette tendance s'observe sur l'ensemble du marche francophone, tous
secteurs confondus.
```

### 3.2 Document 2, le distracteur (identique dans les deux versions)

```
Document 2 :
Etude de cas : optimisation du Site C, agence Nord-Digital. Apres une
refonte technique menee au premier semestre 2026, le Site C affiche un
trafic organique mensuel de 42 000 visites et un taux de conversion de
2,9 %. Le score Core Web Vitals atteint 75/100. Cette progression est
attribuee a une meilleure architecture de liens internes.
```

### 3.3 Document 3, version A (tableau HTML sémantique)

```
Document 3 :
<table>
  <caption>Comparaison des performances SEO de deux sites analysés
  en juin 2026</caption>
  <thead>
    <tr>
      <th scope="col">Indicateur</th>
      <th scope="col">Site A</th>
      <th scope="col">Site B</th>
    </tr>
  </thead>
  <tbody>
    <tr>
      <th scope="row">Pages indexées</th>
      <td>870</td>
```

```
      <td>1 250</td>
    </tr>
    <tr>
      <th scope="row">Trafic organique mensuel</th>
      <td>45 000 visites</td>
      <td>38 000 visites</td>
    </tr>
    <tr>
      <th scope="row">Score Core Web Vitals</th>
      <td>68/100</td>
      <td>92/100</td>
    </tr>
    <tr>
      <th scope="row">Taux de conversion</th>
      <td>3,2 %</td>
      <td>1,8 %</td>
    </tr>
    <tr>
      <th scope="row">Nombre de domaines référents</th>
      <td>340</td>
      <td>180</td>
    </tr>
  </tbody>
</table>
```

## 3.4 Document 3, version B (équivalent en div)

```
Document 3 :
<div class="table">
  <div class="row">
    <div>Indicateur</div>
    <div>Site A</div>
    <div>Site B</div>
  </div>
  <div class="row">
    <div>Pages indexées</div>
    <div>870</div>
    <div>1 250</div>
  </div>
  <div class="row">
    <div>Trafic organique mensuel</div>
    <div>45 000 visites</div>
    <div>38 000 visites</div>
  </div>
  <div class="row">
    <div>Score Core Web Vitals</div>
    <div>68/100</div>
    <div>92/100</div>
  </div>
  <div class="row">
    <div>Taux de conversion</div>
    <div>3,2 %</div>
    <div>1,8 %</div>
  </div>
  <div class="row">
    <div>Nombre de domaines référents</div>
    <div>340</div>
    <div>180</div>
  </div>
</div>
```

### 3.5 Document 4 (identique dans les deux versions)

```
Document 4 :
Interview : les indicateurs a suivre en 2026 selon un consultant SEO
independant. Il est essentiel de suivre regulierement son taux de
conversion et son trafic organique, sans se focaliser uniquement sur
les positions de mots-cles. Un bon suivi passe aussi par l'analyse du
comportement des visiteurs sur la duree.
```

## 4. Les prompts exacts

Chaque prompt est construit selon le même gabarit : `Voici le contenu de plusieurs documents.\n\n{documents}\n\n{question}`. Les quatre documents sont concaténés dans l'ordre 1, 2, 3, 4, séparés par des sauts de ligne. Seule la question varie.

### 4.1 Test 1, comparaison inter-documents avec attribution de source

> Parmi tous les documents fournis, quel site presente le meilleur taux de conversion ? Donne la valeur exacte et indique le numero du document source.

Réponse attendue : Site A, 3,2 %, document 3.

### 4.2 Test 2, question ciblée sur le distracteur

> Quel est le taux de conversion du Site C mentionne dans ces documents ?

Réponse attendue : 2,9 %, document 2.

### 4.3 Test 3, résumé avec attribution de sources

> Resume les taux de conversion de tous les sites mentionnes dans ces documents, avec leur source respective (numero de document).

Réponse attendue : Site A 3,2 % (document 3), Site B 1,8 % (document 3), Site C 2,9 % (document 2).

## 5. Le script de test

Le script ci-dessous correspond à la série de robustesse (3 répétitions, température 0, ordre randomisé). La série initiale utilisait la même architecture d'appels avec une seule exécution par condition et les paramètres de température par défaut de chaque API.

```
import os
import json
import time
import random
import urllib.request
import urllib.error
from datetime import datetime
```

```
from pathlib import Path

ANTHROPIC_API_KEY = os.environ.get("ANTHROPIC_API_KEY")
OPENAI_API_KEY = os.environ.get("OPENAI_API_KEY")
GOOGLE_API_KEY = os.environ.get("GOOGLE_API_KEY")
LLAMA_SERVER_URL = os.environ.get(
    "LLAMA_SERVER_URL", "http://localhost:8085/completion"
)

ANTHROPIC_MODEL = "claude-sonnet-4-6"
OPENAI_MODEL = "gpt-4o"
GOOGLE_MODEL = "gemini-2.5-flash"

RESULTS_DIR = Path("resultats_rag_rep")
RESULTS_DIR.mkdir(exist_ok=True)

REPETITIONS = 3

# Les documents (DOC1, DOC2, DOC3_SEMANTIQUE, DOC3_DIV, DOC4)
# sont definis tels que reproduits en section 3.

VERSIONS = {
    "a": "\n\n".join([DOC1, DOC2, DOC3_SEMANTIQUE, DOC4]),
    "b": "\n\n".join([DOC1, DOC2, DOC3_DIV, DOC4]),
}

PROMPTS = {
    "test1_comparaison_citation": (
        "Parmi tous les documents fournis, quel site presente le meilleur "
        "taux de conversion ? Donne la valeur exacte et indique le numero "
        "du document source."
    ),
    "test2_cible_distracteur": (
        "Quel est le taux de conversion du Site C mentionne dans ces "
        "documents ?"
    ),
    "test3_resume_attribution": (
        "Resume les taux de conversion de tous les sites mentionnes dans "
        "ces documents, avec leur source respective (numero de document)."
    ),
}


def build_prompt(html_content, question):
    return (
        f"Voici le contenu de plusieurs documents.\n\n"
        f"{html_content}\n\n{question}"
    )


def call_with_retry(fn, base_wait=20, max_retries=5):
    for attempt in range(max_retries):
        try:
            return fn()
        except urllib.error.HTTPError as e:
            if e.code == 429 or e.code >= 500:
                wait = base_wait * (attempt + 1)
                time.sleep(wait)
            else:
                return f"ERREUR : HTTP Error {e.code}: {e.reason}"
    return f"ERREUR apres {max_retries} tentatives"


def call_anthropic(prompt):
    def do_call():
        req = urllib.request.Request(
```

```
            "https://api.anthropic.com/v1/messages",
            data=json.dumps({
                "model": ANTHROPIC_MODEL,
                "max_tokens": 1000,
                "temperature": 0,
                "messages": [{"role": "user", "content": prompt}],
            }).encode("utf-8"),
            headers={
                "content-type": "application/json",
                "x-api-key": ANTHROPIC_API_KEY,
                "anthropic-version": "2023-06-01",
            },
            method="POST",
        )
        with urllib.request.urlopen(req) as resp:
            data = json.loads(resp.read())
        return data["content"][0]["text"]
    return call_with_retry(do_call)


def call_openai(prompt):
    def do_call():
        req = urllib.request.Request(
            "https://api.openai.com/v1/chat/completions",
            data=json.dumps({
                "model": OPENAI_MODEL,
                "max_tokens": 1000,
                "temperature": 0,
                "messages": [{"role": "user", "content": prompt}],
            }).encode("utf-8"),
            headers={
                "content-type": "application/json",
                "authorization": f"Bearer {OPENAI_API_KEY}",
            },
            method="POST",
        )
        with urllib.request.urlopen(req) as resp:
            data = json.loads(resp.read())
        return data["choices"][0]["message"]["content"]
    return call_with_retry(do_call)


def call_google(prompt):
    def do_call():
        url = (
            f"https://generativelanguage.googleapis.com/v1beta/models/"
            f"{GOOGLE_MODEL}:generateContent?key={GOOGLE_API_KEY}"
        )
        req = urllib.request.Request(
            url,
            data=json.dumps({
                "contents": [{"parts": [{"text": prompt}]}],
                "generationConfig": {"temperature": 0},
            }).encode("utf-8"),
            headers={"content-type": "application/json"},
            method="POST",
        )
        with urllib.request.urlopen(req) as resp:
            data = json.loads(resp.read())
        return data["candidates"][0]["content"]["parts"][0]["text"]
    return call_with_retry(do_call, base_wait=30)


def call_llama_local(prompt):
    chat_url = LLAMA_SERVER_URL.replace(
        "/completion", "/v1/chat/completions"
```

```
    )
    req = urllib.request.Request(
        chat_url,
        data=json.dumps({
            "messages": [{"role": "user", "content": prompt}],
            "max_tokens": 1500,
            "temperature": 0,
            "repeat_penalty": 1.1,
        }).encode("utf-8"),
        headers={"content-type": "application/json"},
        method="POST",
    )
    try:
        with urllib.request.urlopen(req) as resp:
            data = json.loads(resp.read())
        raw = data["choices"][0]["message"]["content"]
        if "<channel|>" in raw:
            return raw.split("<channel|>", 1)[1].strip()
        return raw.strip()
    except Exception as e:
        return f"ERREUR llama-server : {e}"


MODELS = {
    "claude": call_anthropic,
    "chatgpt": call_openai,
    "gemini": call_google,
    "gemma_local": call_llama_local,
}


def run():
    timestamp = datetime.now().strftime("%Y%m%d-%H%M%S")
    runs = []
    for model_name in MODELS:
        model_runs = []
        for version_name in VERSIONS:
            for prompt_name in PROMPTS:
                for rep in range(1, REPETITIONS + 1):
                    model_runs.append(
                        (model_name, version_name, prompt_name, rep)
                    )
        random.shuffle(model_runs)
        runs.extend(model_runs)

    for model_name, version_name, prompt_name, rep in runs:
        prompt = build_prompt(
            VERSIONS[version_name], PROMPTS[prompt_name]
        )
        response = MODELS[model_name](prompt)
        out_path = RESULTS_DIR / (
            f"{model_name}_v{version_name}_{prompt_name}"
            f"_rep{rep}_{timestamp}.txt"
        )
        out_path.write_text(
            f"Modele : {model_name}\nVersion : {version_name.upper()}\n"
            f"Test : {prompt_name}\nRepetition : {rep}\n"
            f"Date : {timestamp}\n"
            f"Prompt envoye :\n{prompt}\n\nReponse :\n{response}\n",
            encoding="utf-8",
        )
        if model_name == "gemma_local":
            time.sleep(2)
        else:
            time.sleep(8)
```

```
if __name__ == "__main__":
    run()
```

## 6. Les sorties brutes

**Note de transparence :** les réponses reproduites ci-dessous sont issues de la série initiale (une exécution par condition) et ont été récupérées mot pour mot à partir des fichiers sources. La série de robustesse (3 répétitions par condition, température 0) a produit 72 réponses supplémentaires, vérifiées automatiquement par le script (présence des valeurs attendues) puis contrôlées manuellement ; son récapitulatif chiffré figure en section 6.4. Aucune anomalie technique n'a été rencontrée au cours du protocole.

### 6.1 Test 1, comparaison inter-documents avec attribution de source

Réponse attendue : Site A, 3,2 %, document 3. Les 8 réponses de la série initiale sont correctes.

**Claude, version A (tableau sémantique)**

> Le site présentant le meilleur taux de conversion est le Site A, avec un taux de 3,2 %. Source : Document 3 (tableau comparatif des performances SEO de deux sites analysés en juin 2026). À titre de comparaison, le Site B affiche 1,8 % et le Site C (Document 2) affiche 2,9 %, ce qui confirme que le Site A obtient le taux de conversion le plus élevé parmi tous les sites mentionnés dans les documents fournis.

**Claude, version B (div)**

> Le site présentant le meilleur taux de conversion est le Site A, avec un taux de 3,2 %. Source : Document 3 (tableau comparatif Site A / Site B). À titre de comparaison, le Site B affiche 1,8 % et le Site C (Document 2) présente 2,9 %, ce qui confirme que le Site A arrive en tête sur cet indicateur.

**ChatGPT, version A**

> Le site qui présente le meilleur taux de conversion se trouve dans le Document 3. Le Site A affiche un taux de conversion de 3,2 %.

**ChatGPT, version B**

> Le site qui présente le meilleur taux de conversion est le Site A avec un taux de conversion de 3,2 %. Cette information provient du Document 3.

**Gemini, version A**

> Le Site A présente le meilleur taux de conversion. Sa valeur est de 3,2 %. Ce chiffre provient du Document 3.

**Gemini, version B**

> Le site qui présente le meilleur taux de conversion est le Site A avec 3,2 %, selon le Document 3.

**Gemma local, version A**

> Le site qui présente le meilleur taux de conversion est le Site A, avec une valeur de 3,2 % (Document 3).

**Gemma local, version B**

> Le site qui présente le meilleur taux de conversion est le Site A, avec une valeur de 3,2 % (source : Document 3).

### 6.2 Test 2, question ciblée sur le distracteur

Réponse attendue : 2,9 %, document 2. Les 8 réponses de la série initiale sont correctes, aucun modèle ne confond le Site C avec le Site A.

**Claude, version A**

> D'après le Document 2 (étude de cas sur le Site C, agence Nord-Digital), le taux de conversion du Site C est de 2,9 %, mesuré après la refonte technique menée au premier semestre 2026.

**Claude, version B**

> D'après le Document 2, le taux de conversion du Site C est de 2,9 %, tel qu'observé après la refonte technique menée au premier semestre 2026 par l'agence Nord-Digital.

**ChatGPT, version A**

> Le taux de conversion du Site C mentionné dans ces documents est de 2,9 %. Ce détail se trouve dans le Document 2, où il est indiqué que suite à une refonte technique, le Site C affiche un taux de conversion de 2,9 %.

**ChatGPT, version B**

> Le taux de conversion du Site C mentionné dans le Document 2 est de 2,9 %.

**Gemini, version A**

> D'après le Document 2, le taux de conversion du Site C est de 2,9 %.

**Gemini, version B**

> Le taux de conversion du Site C est de 2,9 %.

**Gemma local, version A**

> Le taux de conversion du Site C est de 2,9 % (mentionné dans le Document 2).

**Gemma local, version B**

> Le taux de conversion du Site C est de 2,9 %, comme indiqué dans le Document 2.

## 6.3 Test 3, résumé avec attribution de sources

Réponse attendue : les trois sites, chacun avec sa valeur et son numéro de document (Site A 3,2 % document 3, Site B 1,8 % document 3, Site C 2,9 % document 2). Les 8 réponses de la série initiale restituent les trois valeurs avec les bonnes sources, sous forme de tableau (Claude), de liste numérotée (ChatGPT) ou de liste à puces (Gemini, Gemma local).

Exemple représentatif (Claude, version A), restitué sous forme de tableau Markdown accompagné de la remarque suivante :

> Site A : 3,2 %, Document 3. Site B : 1,8 %, Document 3. Site C : 2,9 %, Document 2.
> Remarque : Le Document 1 et le Document 4 ne mentionnent aucun taux de conversion associé à un site précis. Le Document 4 cite le taux de conversion comme un indicateur important à suivre, mais sans fournir de valeur chiffrée.

Les quatre modèles signalent spontanément, dans au moins une version, que les documents 1 et 4 ne contiennent aucune valeur chiffrée rattachée à un site, ce qui confirme une lecture correcte de l'ensemble du contexte et non une extraction limitée aux seuls passages chiffrés.

## 6.4 Série de robustesse (3 répétitions, température 0)

**Récapitulatif des 72 réponses de la série de robustesse**

| Modèle | Version A (3 questions × 3 répétitions) | Version B (3 questions × 3 répétitions) |
|---|---|---|
| **Claude** | 9/9 correctes | 9/9 correctes |
| **ChatGPT** | 9/9 correctes | 9/9 correctes |
| **Gemini** | 9/9 correctes | 9/9 correctes |
| **Gemma local** | 9/9 correctes | 9/9 correctes |

Sur les 72 réponses, chaque valeur attendue est présente et attribuée au bon document. Aucune confusion entre le Site A (3,2 %, document 3) et le distracteur Site C (2,9 %, document 2) n'a été observée, dans aucune répétition, sur aucune version.

## 7. Synthèse des résultats

**Synthèse consolidée des deux séries**

| Série | Modèle | Version | Question | Résultat | Remarque |
|---|---|---|---|---|---|
| **Initiale** | 4 modèles | A / B | Comparaison avec attribution | Correct / Correct | Site A, 3,2 %, document 3 sur les 8 réponses |
| **Initiale** | 4 modèles | A / B | Question ciblée distracteur | Correct / Correct | 2,9 %, document 2, aucune confusion avec le Site A |
| **Initiale** | 4 modèles | A / B | Résumé avec attribution | Correct / Correct | 3 sites, 3 valeurs, 3 sources exactes sur les 8 réponses |
| **Robustesse ×3** | 4 modèles | A / B | 3 questions × 3 répétitions | 72/72 correctes | Température 0, ordre randomisé, vérification automatique puis contrôle manuel |

### Conclusion consolidée

Sur l'ensemble des 96 réponses collectées (24 en série initiale, 72 en série de robustesse), aucune erreur d'attribution de source n'a été observée, quelle que soit la structure HTML du document contenant les données cibles (tableau sémantique ou équivalent en div). La présence d'un document distracteur contenant la valeur la plus proche du jeu de données (2,9 % contre 3,2 %) n'a créé aucune confusion, sur aucun modèle, dans aucune répétition.

Ce résultat porte exclusivement sur l'étape de génération à partir d'un contexte déjà constitué. Il ne mesure ni l'extraction du HTML, ni le découpage, ni la récupération de documents, étapes qui précèdent la génération dans un système RAG réel et qui font l'objet du test suivant de la série.

Limite documentée : le document contenant les données cibles occupe la troisième position sur quatre dans toutes les conditions. Un effet de position n'a pas été isolé dans ce protocole.

*Document généré à partir des fichiers de résultats bruts du protocole. Pour toute question sur la reproductibilité de ce test, le code source complet est fourni en section 5.*