SIB Text-Mining group at TREC BioGen 2025

Luc Mottin[1], Anaïs Mottaz[1,2], Alexandre Flament[2], Julien Knafou[1,2], Patrick Ruch[1,2]

[1] SIB, Swiss Institute of Bioinformatics, Geneva, Switzerland
[2] HES-SO, University of Applied Sciences and Arts of Western Switzerland, Geneva, Switzerland

Abstract

In the 2025 TREC Biomedical Generative Retrieval (BioGen) Track, we evaluated approaches for producing evidence-grounded biomedical answers across two tasks: sentence-level grounding (Task A) and citation-attributed answer generation (Task B). Our pipelines combined specialized retrieval components with both open-source and instruction-tuned large language models (LLMs), integrating sparse and dense retrieval, re-ranking, and, in later runs, LLM-based claim decomposition, splitting complex claims into positive and negative subclaims for more precise evidence evaluation. Retrieved evidence was used to guide models such as *Qwen2.5-7B-Instruct* and *Meta-Llama-3-8B* in classifying supporting and contradicting references, with PMIDs selected using round-robin strategies to balance subclaim coverage. For Task B, this approach was extended to produce complete citation-grounded biomedical summaries. Across all runs, we aimed at evaluating the impact of hybrid retrieval, model adaptation, and structured prompting on factual grounding, interpretability, and traceability of LLM outputs.

1. Introduction

The adoption of LLMs in biomedical research and clinical decision-making is transforming how scientific knowledge is accessed, synthesized, and applied. Beyond traditional Information Retrieval (IR), these models now generate structured, contextually relevant responses by drawing on extensive biomedical corpora. However, their expanding generative capabilities raise critical questions about transparency and the verifiability of their outputs. In biomedical contexts, where reliability is paramount, systems must not only produce coherent answers but also provide accountable reasoning backed by independently verifiable evidence.

The TREC 2025 BioGen track offers a structured framework to address these challenges [1]. By focusing on two key tasks, respectively for answer grounding (Task A) and reference attribution (Task B), the track assesses the ability of systems to link generative outputs to specific biomedical literature, capturing both supporting and contradictory evidence. This benchmark bridges traditional IR evaluation with the emerging need to validate the factual accuracy of LLM-generated content, a critical step toward developing trustworthy biomedical Artificial Intelligence (AI).

As part of the Swiss Institute of Bioinformatics (SIB), the SIB Text Mining group participates in both BioGen 2025 tasks, building on our ongoing research into explainable and reliable biomedical language technologies. Drawing from our experience in previous TREC tracks, including Genomics [2], Medical Records [3], Chemical IR [4], Clinical Decision Support [5], Deep Learning [6], Precision Medicine [7], and PLABA [8], we treat the BioGen challenge as an opportunity to systematically evaluate evidence-aware generation strategies. Our work examines the interplay between retrieval precision, citation diversity, and generative factuality, using both open-source and instruction-tuned LLMs to ground generated content in verifiable evidence. Together, these approaches aim to clarify how specialized retrieval components and structured prompting can improve the verifiability and citation quality of biomedical answers.

2. Methodology

2.1. Data and resources

To support our participation in the TREC 2025 BioGen Track, we relied on a robust computing infrastructure enabling the rapid prototyping of multiple retrieval and generation pipelines in parallel. The availability of two high-performance servers (see Table 1), each equipped with multi-core Intel Xeon processors, 1 TB of RAM, and high-memory NVIDIA GPUs (A100 80GB and H100 94GB), allowed us to explore diverse approaches within a short timeframe. This setup proved particularly useful in managing the track's dual tasks concurrently, facilitating iterative experimentation and comparative evaluation of various LLM configurations, retrieval depths, and grounding strategies.

The infrastructure included 163 TB of HDD and 55 TB of SSD storage, ensuring efficient indexing and fast access to large-scale biomedical corpora, primarily derived from SIBiLS/BiodiversityPMC [9], which includes an extended mirror of PubMed Central. System deployment was handled through Proxmox with LVM-Thin virtualization, enabling flexible resource allocation. Automated configuration and orchestration were managed using Ansible, while OPNSense ensured secure network access. All components ran on Ubuntu 24.04, providing compatibility with current machine learning libraries and tools.

Server	Model	CPU	RAM	GPU
Hulk	DELL PowerEdge R750XA	2 x Intel Xeon Gold 6330	1 TB	2 x A100 80GB
Yoda	DELL PowerEdge R750XA	2 x Intel Xeon Gold 6548N	1 TB	2 x H100 94GB

Table 1. Server specifications used for experimentation.

Regarding the data, we rely on the PubMed 2025 baseline as the primary document corpus for IR. The corpus is preprocessed and made accessible through the official distribution, and the accompanying starter kit provides automated tools for downloading and indexing the collection. For system development, we utilize question-answer pairs and {topic + question + narrative} triplets as inputs for Task A and B, respectively. The BioGen 2024 assessments were also supplied as a benchmark resource for tuning and validating the models.

2.2. Task A: Grounding Answer

This year, Task A of TREC BioGen focuses on the identification of relevant biomedical literature to support or challenge individual sentences within an answer to a given biomedical question. Using a corpus of PubMed articles, systems must retrieve PMIDs per sentence, prioritizing those that contradict the statement, followed by those that support it. Each sentence is initially paired with outdated supporting references, and the goal is to supplement these with more pertinent citations from the provided corpus.

Run a-1 - baseline

In this run, a lightweight retrieval and citation grounding pipeline was implemented, leveraging Lucene for IR and SpaCy-based semantic similarity for evidence classification. For each biomedical claim, Lucene was used to perform keyword-based retrieval over the provided PubMed corpus, ranking documents using a standard TF-IDF scoring schema. Candidate abstracts retrieved by Lucene were then analyzed using SpaCy's *en_core_sci_lg* model to compute semantic similarity between the claim and the retrieved documents [10]. Based on similarity scores, documents were classified and the top three were selected as supporting citations. Contradicting citations were not considered in this run as the model is designed to estimate general contextual alignment (not to distinguish refuting from supporting evidence). Finally, this approach prioritizes computational efficiency and provides a baseline for our subsequent runs.

Run a-2

The 2nd run extends the baseline by replacing the semantic similarity component with a textual entailment model, specifically the *bart-large-mnli* [11]. As in run a-1, candidate documents are retrieved for each biomedical claim using Lucene. However, instead of relying on lexical similarity, the model evaluates whether the retrieved abstracts' content entails, contradicts, or is neutral toward each claim. This model captures deeper semantic and logical relations, making it more robust to paraphrased or indirect evidence, a common feature in biomedical writing. A strict prompt was introduced to guide the model more effectively in distinguishing supporting from contradicting evidence, drawing on prompt engineering techniques from the biomedical domain [12,13].

```
You are classifying whether a scientific document agrees with, disagrees with, or is irrelevant to a claim. \n
Claim: '{claim}' \n
Question context: '{question}' \n
Instructions: \n
1- Choose 'contradicts' ONLY if the document explicitly provides evidence that conflicts with the claim. \n
2- Choose 'supports' ONLY if the document explicitly provides evidence that aligns with the claim. \n
3- Choose 'irrelevant' if the document does not clearly support or contradict the claim, or if it only discusses the topic indirectly. \n
4- Be strict: do not infer beyond what is stated. \n
Documents {{}}
```

Prompt used to classify supporting and contradicting documents in runs a-2 and a-5.

Run a-3

This run combines sparse retrieval using ElasticSearch (ES) with dense retrieval via FAISS [14], using embeddings from *MedCPT-Query-Encoder* [15]. Retrieved candidates are then reranked using the *bge-reranker-v2-m3* model to improve precision [16]. Finally, the top-ranked documents are passed to *BioMistral-7B-SLERP* [17], prompted[1] to identify supporting and contradicting citations. Compared to runs a-1 and a-2, this pipeline aims to enhance retrieval coverage and grounding accuracy by leveraging hybrid search and LLM-based evidence interpretation.

Run a-4

The aim of this run is to assess the impact of switching the language model used for grounding. Here, we replicated the retrieval pipeline from run a-3, combining ES, FAISS-based dense retrieval, and reranking with *bge-reranker-v2-m3*, but replaced the classification model with *Bio-Medical-Llama-3-8B* [18]. This substitution allows us to evaluate whether a larger model provides improved clinical reasoning and if it enhances precision in downstream classification.

Run a-5

This run builds on the SIBiLS[2] retrieval model for initial article discovery [19], targeting a higher recall in initial article retrieval and feeding a richer set of documents into the classification stage. Consistent with run a-2, the retrieved documents are classified as supporting or contradicting using the *bart-large-mnli* model, guided by the same prompt.

Run a-6

Here, we developed a pipeline that decomposes each biomedical claim into positive subclaims and their corresponding negations. This LLM-driven claim pre-processing step was introduced to provide the model with explicit, testable propositions rather than single compound statements.

```
Task:
You are given a CLAIM. Your job is to return EXACTLY A JSON array of sub-claims (strings) if possible.
Each sub-claim must express a single, independent element of the original claim.
- Try splitting in different sentences as much as possible, keeping the meaning
- Try to re-create a correct sentence
```

[1] https://gist.github.com/dalf/0146d0d5e09fe7f76dd2e468d4f6ba2b

[2] https://sibils.org/

```
- If the CLAIM is short or cannot be decomposed, return a JSON array with the CLAIM itself as the only element.
- Do not include any commentary, labels, or extra text—return ONLY the JSON array.

IMPORTANT:
- The content inside <EXAMPLE> is ONLY an illustration. Do NOT use it as input.
- Work ONLY on the CLAIM provided below inside <INPUT>.
- Output MUST be valid JSON (an array of strings). No trailing commas. No extra text.

 <EXAMPLE>
 CLAIM:
 "Sleep apnea can be prevented by losing weight, quitting alcohol, and changing sleep position."

 EXPECTED OUTPUT:
 [
   "Sleep apnea can be prevented by losing weight.",
   "Sleep apnea can be prevented by quitting alcohol.",
   "Sleep apnea can be prevented by changing sleep position."
 ]
 </EXAMPLE>

<INPUT>
{claim}
</INPUT>

Return ONLY the JSON array on a single line.
```

Prompt used to generate subclaims in runs a-6, a-7 and b-4.

```
Task:
You are given a CLAIM. Your job is to return the opposite claim as EXACTLY ONE JSON string.
- Do not include any commentary, labels, or extra text—return ONLY the JSON array.

<EXAMPLE>
CLAIM: "Sleep apnea can be prevented by losing weight."
EXPECTED OUTPUT: "Sleep apnea cannot be prevented by losing weight."
</EXAMPLE>

IMPORTANT:
- The content inside <EXAMPLE> is ONLY an illustration. Do NOT use it as input.
- Work ONLY on the CLAIM provided below inside <INPUT>.
- Output MUST be valid JSON a string. No trailing commas. No extra text.

<INPUT>
{claim}
</INPUT>

Return ONLY the JSON array on a single line.
```

Prompt used to generate negative subclaims in runs a-6, a-7 and b-4.

Each subclaim is used as a query for k-Nearest Neighbors (kNN) retrieval in ES, leveraging dense embeddings from the *S-PubMedBert-MS-MARCO* model [20]. These embeddings are reduced via Principal

Component Analysis (PCA) and combined with BM25 scores (weight = 0.3), with a similarity threshold set at 0.7. Retrieved evidence is then processed by the *Qwen2.5-7B-Instruct* language model [21], which is prompted to identify both supporting citations (*i.e.*, publications confirming positive subclaims) and contradicting citations (*i.e.*, publications confirming the negated subclaims), as presented below. From these outputs, up to three PMIDs per category are selected based on the initial retrieval scores. When available, truncated abstracts are expanded to full text using SIBiLS to enhance prompt input, though this extended content is not used for retrieval.

```
You are a careful biomedical reviewer.
Task:
Given a QUESTION, a candidate CLAIM, and EVIDENCE as full PubMed abstracts, your job is to return EXACTLY A JSON object with the key "confirm_pmids", whose value is a LIST of PubMed IDs (integers). These IDs must correspond to abstracts that STRICTLY confirm the CLAIM at the level of the sentence.

IMPORTANT:
- Consider especially the title (=FIRST SENTENCE) and conclusion (LAST SENTENCES) of the abstract
- "Strictly confirm" = directly supports the CLAIM (not speculative; population, context, and outcome must align)
- Check that the abstract contains a sentence CONFIRMING the CLAIM (e.g. common keywords or synonyms, same direction of relation)
- Use careful reasoning on the abstracts: the relationship’s direction can be ambiguous
- Work ONLY on the CLAIM and EVIDENCE provided below inside <INPUT>
- Output MUST be valid JSON. No trailing commas. No extra text

<INPUT>
CLAIM: {claim}
EVIDENCE: {evidence}
</INPUT>

Return ONLY the JSON object.
```

Prompt used to identify the supporting citations (for positive & negative claims) in runs a-6, a-7 and b-4.

Run a-7

Building on the pipeline from run a-6, each biomedical claim is decomposed into positive subclaims and their corresponding negations, followed by kNN retrieval using dense *S-PubMedBert-MS-MARCO* embeddings (reduced via PCA) combined with BM25 scoring (weight = 0.3) and a similarity threshold of 0.7. Retrieved evidence is evaluated by the *Qwen2.5-7B-Instruct* model, prompted to identify supporting and contradicting citations, as in run a-6. In this 7th run, the difference with run a-6 is the use of a round-robin selection strategy that cycles through the publications retrieved for each subclaim to select up to three PMIDs, ensuring a more balanced coverage across all subclaims by considering both subclaim representation and retrieval scores [22]. As in the previous run, truncated abstracts are expanded to full text via SIBiLS to enhance the prompt.

2.3. Task B

Task B builds on the grounding principles of Task A by requiring systems to generate complete biomedical answers with embedded citation for each sentence. Using the same PubMed corpus, the task emphasizes precise reference attribution, ensuring that every statement in the output is directly supported by relevant publications. This task evaluates the integration of factual generation with evidence-based sourcing and, while mirroring certain aspects of Retrieval-Augmented Generation (RAG) workflows, relies entirely on provided data rather than dynamic retrieval [23]. This distinction highlights the importance of careful selection and curation of the input corpus to ensure reliability and accuracy.

Run b-1

To establish a baseline, we implemented a pipeline that combines biomedical literature retrieval with controlled LLM generation. Specifically, we employed SIBiLS as the retrieval backbone to identify relevant biomedical documents and evidence passages associated with each query. Retrieved items were then incorporated into carefully designed prompt templates aimed at maximizing factual grounding and contextual coherence. The enriched prompts were subsequently processed by *Meta-Llama-3-8B*, generating candidate answers conditioned on the retrieved context. The goal of this approach is to enhance both the accuracy and traceability of generated biomedical responses.

```
You are a biomedical scientific assistant.\n
 Your task is to answer the question using ONLY the provided PubMed documents, strictly following these rules:\n\n
1. **Exclusive Source Use**: Base your answer solely on the provided documents. Never use external knowledge or unsupported inferences.\n
2. **Structure and Style**:\n
  - The response MUST form a coherent paragraph.\n
  - Be simple, factual and avoid speculation or interpretation not directly supported by the documents.\n
3. **Reference Attribution**:\n
  - Each sentence MUST cite between 1 and 3 references\n
  - Insert references immediately after the sentence, formatted as: "[1],[2]."\n
  - **Direct Support Rule**: Every PMID cited must directly support the claim made in its corresponding sentence. Never cite a PMID unless the document explicitly contains the information stated in that sentence.\n
  - Do NOT include a Bibliography or Reference list at the end of the paragraph.\n
4. **Constraints**:\n
  - Do NOT repeat the Question or Context\n
  - Maximum length: 250 words\n
5. **Example of Expected Format**:\n
"The mechanism of action of _molecule_ involves competitive inhibition of _target_ [1],[3]. This interaction was confirmed in vitro in human cell lines [2]."\n\n
--------------------\n
f"Context: {narrative}\n
--------------------\n
f"Question: {query}\n
--------------------\n
Documents: {{}}\n
```

Prompt used in runs b-1, b-2 and b-3 to answer the biomedical questions based on the retrieved citations.

Run b-2

Building on the baseline pipeline from run b-1, this approach retains the SIBiLS-based retrieval to provide relevant biomedical literature but introduces a key modification in the generation stage. Here, the *Meta-Llama-3-8B* model is fine-tuned using LoRA (Low-Rank Adaptation) to efficiently adapt the model to this specific task using BioGen 2024 data while keeping computational costs low [24].

Run b-3

In this run, the same SIBiLS-based retrieval pipeline is retained, while the generative component is replaced with *Qwen2.5-7B-Instruct*. Retrieved evidence from SIBiLS is integrated into the model input to guide answer generation, testing its ability to produce more context-aware and evidence-grounded outputs in this end-to-end setup.

Run b-4

Building on the approaches from runs a-6 and a-7 (Task A), this pipeline adds a step of claim generation from patient's history before retrieval and selection of evidence, followed by the synthesis of found evidence. The *Qwen2.5-7B-Instruct* model first generates explicit biomedical claims derived from questions based on patient history as presented in the prompt below.

```
You are a biomedical assistant.
Task:
Given a patient's HISTORY, do the following:
1. Imagine several specific QUESTIONS that could naturally arise from this history (but do not output them).
2. For each question, generate a clear, atomic, affirmative CLAIM that could serve as a possible answer.
   - Each claim must be a single, testable statement.
   - Claims must be written as factual affirmations, not as questions.
   - Avoid duplicates, trivial rephrasings, or overly broad/general statements.

Output format:
- Return ONLY a JSON array of strings (claims).
- Do not include explanations, commentary, or any extra text.

<EXAMPLE>
HISTORY:
"During an ED visit after a car accident, the patient was told an axonal injury was detected. The patient is concerned it will affect his health."

EXPECTED OUTPUT:
[
  "Diffuse axonal injury is a type of brain injury caused by rapid acceleration and deceleration forces.",
  "Diffuse axonal injury can lead to long-term cognitive and physical symptoms.",
  "MRI scans can detect evidence of diffuse axonal injury in the brain."
]
</EXAMPLE>

<INPUT>
{history}
</INPUT>
```

Prompt used to create the claims from patient's history in run b-4.

Similarly to what was done in task a-6 and a-7, each generated claim is then decomposed into positive subclaims and their negations (see prompts from Run a-6). For each positive and negative subclaim, kNN retrieval is performed in ES using embeddings from *S-PubMedBert-MS-MARCO* reduced via PCA and combined with BM25 scoring (weight = 0.3, threshold = 0.7). Retrieved PMIDs are provided to *Qwen2.5-7B-Instruct*, which categorizes references as supporting citations for the positive and negative subclaims (see prompt from Run a-6). Up to three PMIDs per positive and negative claims are selected using a round-robin strategy to ensure coverage across subclaims. Finally, the model generates a summary integrating the claims and their associated references into a patient-friendly summary as presented below.

```
You are a concise biomedical writer.
Task: Given a QUESTION and EVIDENCE SENTENCES (each tagged as [PMID:NNNNN|SENT:i]),
produce 3–8 short, patient-friendly statements. For each statement, list the supporting PMIDs.
```

```
Rules:
- Return ONLY a JSON object with key "responses": list of objects:
  {{ "text": str, "citations": [int, ...] }}
- Use ONLY PMIDs present in the evidence.
- Combine sentences when helpful; keep neutral, factual tone.

QUESTION: {question}

EVIDENCE SENTENCES: {evidence}
```

Prompt used to generate the patient-friendly summary in run b-4.

3. Results

Table 2 reports the automatic evaluation results for Task A. It includes the evaluation of all submitted runs using the BioACE framework [25], based on the Llama-3-70B model, and reports precision, recall, and F1-scores for supported and contradicted citations at the claim level.

Tables 3 and 4 present the results for Task B. Table 3 includes automatic evaluation results for all runs using BioACE to report on Nugget Precision, Nugget Recall, Completeness, and Correctness. Table 4 focuses on citation evaluation, reporting Citation Coverage, Citation Support Rate, and Citation Contradiction Rate, based on model-predicted attributions for each answer sentence.

run	supported precision	supported recall	supported F1	contradicted precision	contradicted recall	contradicted F1
a-1	52,41	74,23	58,87	0	0	0
a-2	47,85	64,09	52,3	3,69	5,67	4,21
a-3	15,12	26,8	18,42	0	0	0
a-4	17,01	32,47	21,31	0	0	0
a-5	46,22	64	51,17	2,06	3,95	2,56
a-6	36,25	42,14	37,82	1,98	2,58	2,15
a-7	39,52	45,02	41,08	2,06	2,58	2,23

Table 2. Automatic evaluation results for Task A.

run	precision	recall	completeness	correctness
b-1	92,32	35,8	28,79	63,93
b-2	90,29	33,28	54,3	64,19
b-3	85,74	33,81	38,98	66,32
b-4	85,1	30,52	72	61,67

Table 3. Automatic evaluation results for Task B - all.

run	citation coverage	citation support rate	citation contradict rate
b-1	48,39	66,83	2,48
b-2	44,71	77,78	1,23
b-3	98,21	95,39	0,66
b-4	78,08	86,9	1,19

Table 4. Automatic evaluation results for Task B - citation.

4. Discussion

Regarding sentence-level grounding (Task A), we hypothesized that combining sparse and dense retrieval methods would provide a more robust strategy to identify relevant evidence. Sparse methods such as BM25 were expected to perform well for exact term matching, while dense retrieval using kNN search in ES with

S-PubMedBert-MS-MARCO or *MedCPT-Query-Encoder* with FAISS was anticipated to improve generalization across paraphrased or semantically similar texts. However, the results indicate that increased architectural complexity did not systematically translate into improved performance. Based on a TF-IDF retrieval and SpaCy semantic similarity classification, run a-1 achieved the highest supported recall (74.23%) and supported F1 (58.87%). On the other hand, we also evaluated entailment and generative models for evidence classification, including *bart-large-mnli*, *Qwen2.5-7B-Instruct*, *BioMistral-7B-SLERP* and *Bio-Medical-Llama-3-8B*, which were expected to better capture contextual nuances and perform fine-grained reasoning about claim-evidence relations. While these models offer richer contextual understanding, their advantage was not clearly reflected in overall supported F1 scores.

In runs a-6 and a-7, we introduced an LLM-based claim pre-processing step to enhance contradiction detection. Each complex claim was decomposed into distinct subclaims, and explicit negative counterparts were generated for each of them. This procedure transformed a single compound statement into a set of clearly defined, testable propositions intended to guide both retrieval and classification. The motivation was to reduce ambiguity in contradiction prompts, since a claim such as "X is due to Y" may be contradicted either by a direct negation ("X is not due to Y") or by an alternative explanation ("X is due to Z"), which standard prompts may conflate. Despite this structured approach, contradiction detection remained particularly challenging across all configurations. This limited performance is likely not solely attributable to the classification models, but rather to the retrieval setup itself: queries derived directly from the original claim primarily retrieve semantically similar documents, whereas contradictory evidence may be expressed through different conceptual framings or alternative causal attributions. Retrieval mechanisms optimized for similarity are therefore ill-suited to systematically capture semantically opposing evidence, which may explain the consistently low contradiction scores.

For citation grounded generation (Task B), all runs achieved very high precision (85-92%), indicating that generated answers rarely introduced unsupported content; run b-1 obtained the highest precision (92.3%), while run b-4 achieved the highest completeness score (72%). Citation-level evaluation revealed clearer differences between configurations. Run b-3 obtained the strongest citation grounding, with very high citation coverage (98.21%) and support rate (95.39%), but the lowest contradiction rate (0.66%). Run b-4 also performed strongly (coverage 78.08%, support rate 86.9%, contradict rate 1.19%), building on the claim-based methodology developed in Task A. These results indicate that integrating structured retrieval with guided generation not only strengthens factual grounding but also enhances the transparency and traceability of the resulting biomedical summaries.

5. Conclusion

Across all runs in Tasks A and B, we explored progressively structured ways of combining retrieval and generation to improve factual grounding. This track is highly relevant in biomedical research, where AI tools are increasingly used to generate scientific claims. Being able to detect not only relevant but also reliable and contradictory sources is essential to maintain the rigor of analyses and prevent the spread of misinformation. Our results suggest that our pipelines facilitate explicit reasoning steps and traceable links between retrieved evidence and generated output, as opposed to end-to-end generative or less structured RAG approaches. However, further efforts are required to enhance the detection of contradictory evidences.

Bibliography

1. Gupta D, Demner-Fushman D, Hersh W et al. (2024) Overview of TREC 2025 Biomedical Generative Retrieval (BioGen) Track. In The Thirty-Fourth Text REtrieval Conference Proceedings (TREC2025). NIST Special Publication.
2. Ruch P, Chichester C, Cohen G et al. (2004). Report on the TREC 2003 Experiment: Genomic Track. In Proceedings of the 2003 Text REtrieval Conference (TREC).
3. Gobeill J, Gaudinat A, Pasche E et al. (2011) Bitem group report for TREC medical records track 2011. In the Proceedings of the Twentieth text retrieval conference (TREC 2011). NIST Special Publication.

4. Gobeill J, Gaudinat A, Pasche E et al. (2011) Bitem group report for TREC chemical IR track 2011. In the Proceedings of the Twentieth text retrieval conference (TREC 2011). NIST Special Publication.
5. Gobeill J, Gaudinat A et Ruch P (2015) Exploiting incoming and outgoing citations for improving Information Retrieval in the TREC 2015 Clinical Decision Support Track. In Proceedings of The 24th Text REtrieval Conference (TREC 2015). NIST Special Publication.
6. Knafou J, Jeffreys M., Mottin L et al. (2019) SIB Text Mining at TREC 2019 Deep Learning Track: Working Note. In Proceedings of the Twenty-Eighth Text REtrieval Conference (TREC 2019). Gaithersburg (Maryland). NIST Special Publication.
7. Pasche E, Caucheteur D, Mottin L et al. (2020) SIB text mining at TREC precision medicine 2020. In Proceedings of the Twenty-Ninth Text REtrieval Conference (TREC 2020). Gaithersburg (Maryland). NIST Special Publication. NIST Special Publication.
8. Mottin L, Mottaz A, Knafou J et al. (2024) SIB Text-Mining at TREC PLABA 2024. In Proceedings of the Thirty-Third Text REtrieval Conference (TREC 2024). Gaithersburg (Maryland). NIST Special Publication, NIST SP1329.
9. Pasche E, Gobeill J, Agosti D et al. (2023) From SIBiLS to Biodiversity PMC: Foundations for the One Health Library. Biodiversity Information Science and Standards, 7, e111660. DOI: 10.3897/biss.7.111660.
10. Honnibal M., Montani I., Van Landeghem S. et al. (2020). spaCy: Industrial-strength Natural Language Processing in Python. DOI: 10.5281/zenodo.1212303.
11. Lewis M, Liu Y, Goyal N et al. (2019) BART: Denoising Sequence-to-Sequence Pre-training for Natural Language Generation, Translation, and Comprehension. arXiv eprint, 1910.13461. DOI: 10.48550/arXiv.1910.13461.
12. Meskó B. (2023) Prompt Engineering as an Important Emerging Skill for Medical Professionals: Tutorial. Journal of Medical Internet Research, 25, e50638. DOI: 10.2196/50638.
13. Wang L, Chen X, Deng X et al. (2024) Prompt engineering in consistency and reliability with the evidence-based guideline for LLMs. NPJ Digital Medicine, 7(1), 41. DOI: 10.1038/s41746-024-01029-4.
14. Douze M, Guzhva A, Deng C et al. (2025) The Faiss library. arXiv eprint, 2401.08281. DOI: 10.48550/arXiv.2401.08281.
15. Jin Q, Kim W, Chen Q, et al. (2023) MedCPT: Contrastive Pre-trained Transformers with large-scale PubMed search logs for zero-shot biomedical information retrieval. Bioinformatics, 39(11), btad651. DOI: 10.1093/bioinformatics/btad651.
16. Chen J, Xiao S, Zhang P et al. (2024) BGE M3-Embedding: Multi-Lingual, Multi-Functionality, Multi-Granularity Text Embeddings Through Self-Knowledge Distillation. arXiv eprint, 2402.03216. DOI: 10.48550/arXiv.2402.03216.
17. Labrak Y, Bazoge A, Morin E et al. (2024) BioMistral: A Collection of Open-Source Pretrained Large Language Models for Medical Domains. arXiv eprint, 2402.10373. DOI: 10.48550/arXiv.2402.10373.
18. ContactDoctor (2024) ContactDoctor-Bio-Medical: A High-Performance Biomedical Language Model. Published on https://huggingface.co/ContactDoctor/Bio-Medical-Llama-3-8B.
19. Gobeill J, Caucheteur D, Michel PA et al. (2020) SIB Literature Services: RESTful customizable search engines in biomedical literature, enriched with automatically mapped biomedical concepts, Nucleic Acids Research, 48(W1), W12–W16. DOI: 10.1093/nar/gkaa328.
20. Deka P, Jurek-Loughrey A et Padmanabhan D. (2022) Improved methods to aid unsupervised evidence-based fact checking for online health news. Journal of Data Intelligence, 3(4), 474–505. DOI: 10.26421/JDI3.4-5.
21. Yang A, Yang B, Hui B et al. (2025) Qwen2.5 Technical Report. arXiv eprint, 2412.15115. DOI: 10.48550/arXiv.2412.15115.
22. Fürnkranz J (2002) Round robin classification. Journal of Machine Learning Research, Volume 2, 721-747. DOI: 10.1162/15324430232088460.
23. Koga S, Ono D et Obstfeld A. (2025) Retrieval-augmented generation versus document-grounded generation: a key distinction in large language models. Journal of Pathology: Clinical Research, 11(1), e70014. DOI: 10.1002/2056-4538.70014.
24. Hu EJ, Shen Y, Wallis P et al. (2021) LoRA: Low-Rank Adaptation of Large Language Models. arXiv eprint, 2106.09685. DOI: 10.48550/arXiv.2106.09685.
25. Gupta D, Bartels D et Demner-Fushman D (2026) BioACE: An Automated Framework for Biomedical Answer and Citation Evaluations. arXiv eprint, 2602.04982. DOI: 10.48550/arXiv.2602.04982.