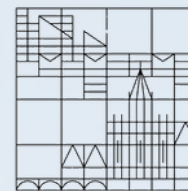


10 | Fine-Tuning LLMs

Giordano De Marzo

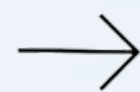
<https://giordano-demarzo.github.io/>

Deep Learning for the Social Sciences



Masked Language Modeling

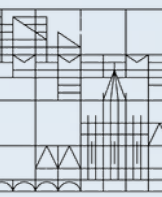
But I do
think it is
their husbands'
faults if
wives do fall.



But I do
[dance] it is
their [MASK]
faults if
[MASK] do fall.

Masked Language Modeling (MLM) is the breakthrough idea that made BERT possible. This becomes the primary training objective for encoder-only models.

- **Random masking strategy:** Replace around 15% of tokens during training
- **Masking breakdown:** 80% → [MASK], 10% → random word, 10% → unchanged
- **Training objective:** Cross-entropy loss on masked token predictions
- **Self-supervised learning:** No labeled data needed, just raw text



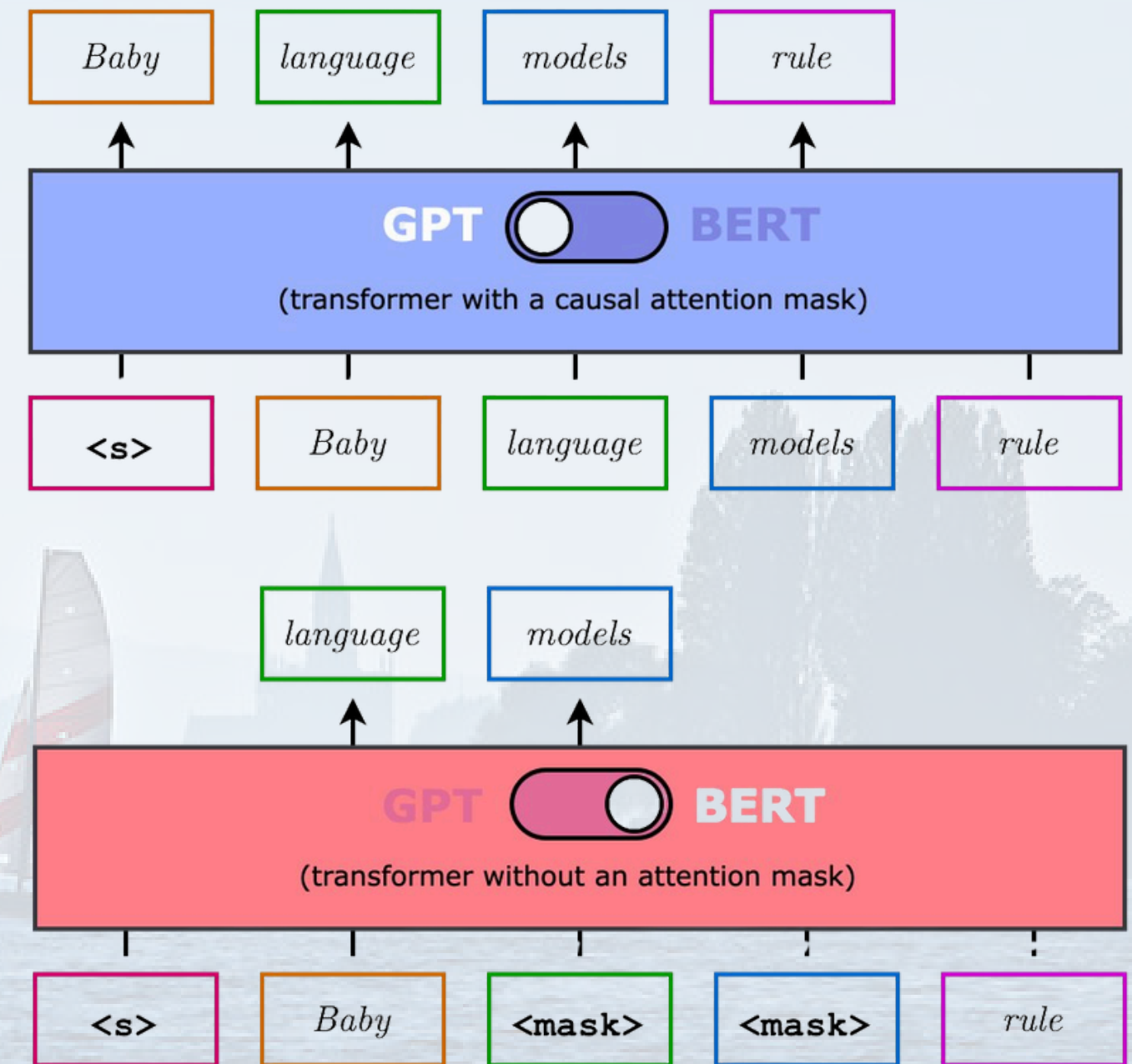
Next Token Prediction

Decoder-only models have a simple training objective: given some text, predict what comes next.

- **Training setup:** Take any text sequence, predict each next token
- **Loss function:** Cross-entropy loss on next token predictions
- **Self-supervised:** No human labels needed
- **Scalable:** Any text from internet can be training data

Next token prediction requires understanding grammar, facts, reasoning, and context to predict well.

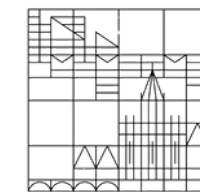
This simple objective leads to emergent capabilities like reasoning or few-shot learning





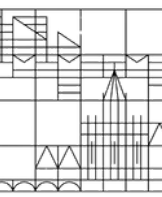
Outline

1. Inference with LLMs
2. Intro to Fine-Tuning
3. Forecasting Radical Innovation
4. Pollution Abatement Technologies



Inference with LLMs

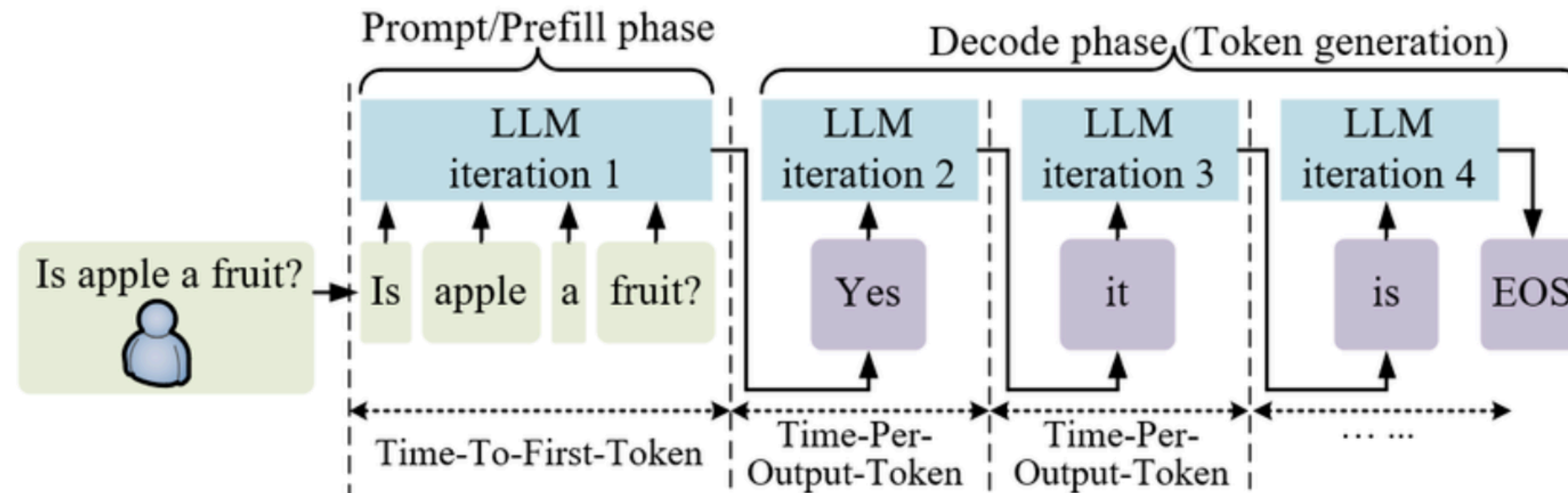


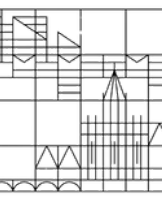


Inference with LLMs

LLM inference is the process of using a pre-trained language model to generate responses or predictions without updating its parameters.

- **Input:** Natural language prompts or queries
- **Process:** Forward pass through the model's neural network
- **Output:** Generated text, completions, or structured responses

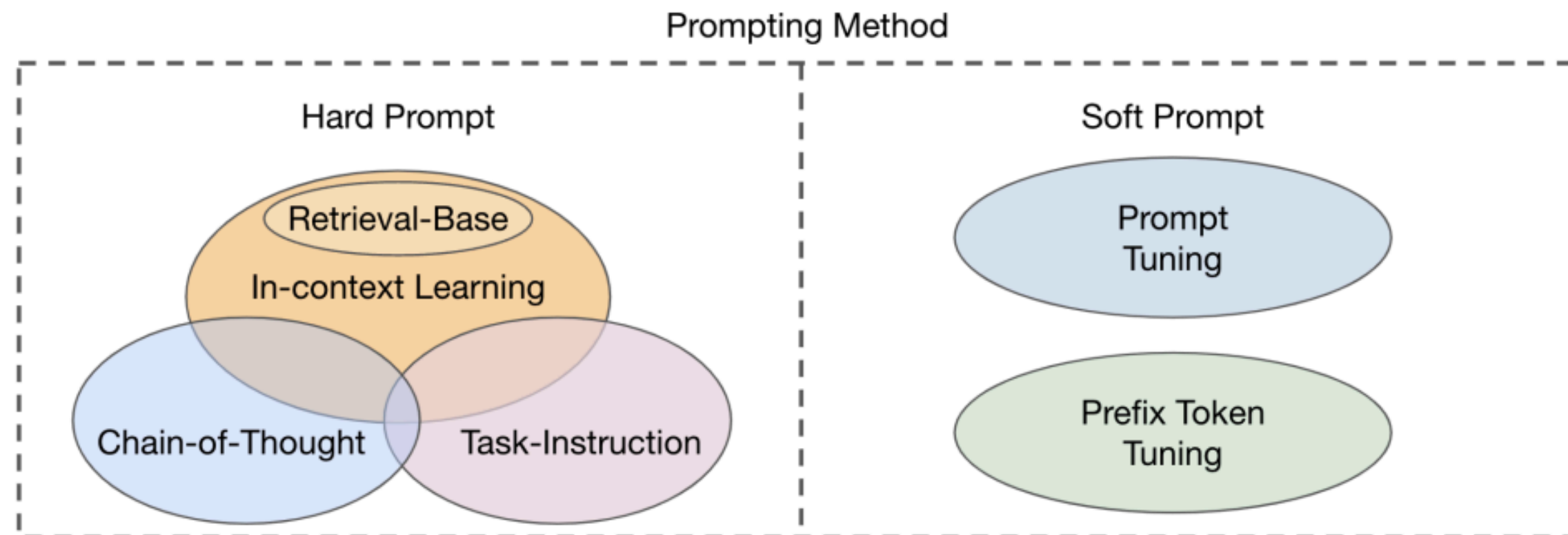


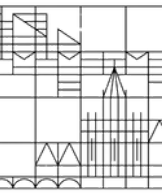


Prompt Engineering

Prompt engineering is a crucial technique in working with large language models, enabling users to obtain better and more accurate outputs. The main approaches to prompt engineering include:

- **Few-Shot Learning:** Providing the model with a few examples in the prompt
- **Chain of Thought:** Structuring prompts to guide the model through a logical sequence
- **Soft Prompts:** Using learnable prompt tokens that adapt during training





Zero-Shot vs Few-Shot

In Few-Shot learning the model is provided with a few examples within the prompt

- Different from Zero-Shot learning where no examples are provided
- This is an example of in-context learning
- This helps the model understand the task better and produce more accurate responses.
- Typically up to 3-5 examples can be useful, if this does not work, fine tuning should be considered



You are an AI assistant who can decode emotion analysis.

Example 1:

This movie is great, I had a great time watching it.

Result 1:

Positive

Example 2:

I've never seen a worse movie, it was a waste of time.

Result 2:

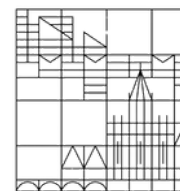
Negative

Example 3:

The food is bad and the service should be improved



Negative



Chain of Thoughts

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. $5 + 6 = 11$. The answer is 11.

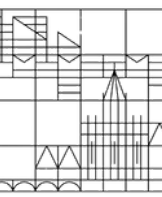
Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had $23 - 20 = 3$. They bought 6 more apples, so they have $3 + 6 = 9$. The answer is 9. ✓

Chain of Thought (CoT) prompting is a technique used to guide large language models through a logical sequence of steps to arrive at a solution.

- This approach helps improve the model's reasoning abilities by breaking down complex tasks into simpler, manageable parts.
- CoT involves structuring the prompt to include intermediate steps and reasoning processes.
- Example: For a math problem, instead of directly asking for the answer, the prompt asks for the steps to solve the problem, leading the model to a logical conclusion.



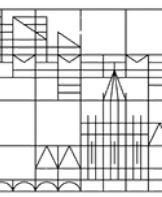
Closed Models

Closed LLMs are proprietary models where the weights, architecture, and training data remain private to the company that developed them.

- OpenAI's GPT-4, Anthropic's Claude, Google's Gemini
- State-of-the-art performance, extensive safety training
- Access only through company-controlled interfaces
 - **Advantages:** High quality, reliable infrastructure, regular updates
 - **Limitations:** No model access, dependency on provider, ongoing costs

These models represent the current frontier of LLM capabilities but require trust in the provider.





Web Interface vs API

In cosa posso essere utile?

A screenshot of a chat interface input field. The text "Fai una domanda" is visible in the input area. Below the input field, there is a plus sign, a gear icon, and the text "Strumenti". To the right of the input field, there is a microphone icon and a mute icon.

```
python ↕
1 from openai import OpenAI
2 client = OpenAI()
3
4 response = client.responses.create(
5     model="gpt-4.1",
6     input="Write a one-sentence bedtime story about a unicorn."
7 )
8
9 print(response.output_text)
```

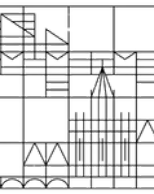
Closed models can be accessed through user-friendly web interfaces or programmatic APIs.

Web Interfaces:

- ChatGPT, Claude, Gemini web applications
- Conversation memory, file uploads, safety filters
- Mostly used for general-purpose tasks

API Access:

- REST APIs for building applications
- Control on temperature, max tokens, system prompts, response formatting

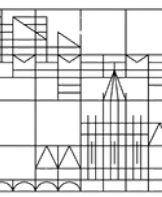


Open Models

“Open” models come in different flavors with varying degrees of transparency and accessibility.

- **Open weights:** Model weights released
- **Open source:** Complete transparency including training code and methodology (Pythia, OLMo)
- **Open data:** Training datasets publicly available for reproducibility (C4, RedPajama)
- **Fully open:** Weights + source code + data + training process (rare but ideal)

Most “open” models are actually just “open weights” models

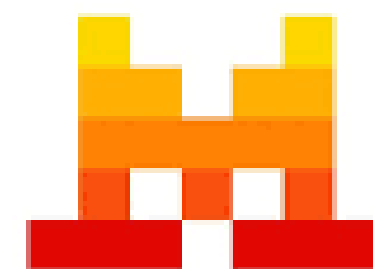


SOTA Open Models

The open model landscape features strong competitors from major tech companies and specialized AI labs.

- **Meta:** Llama 4 series
- **Alibaba:** Qwen 3.6 series
- **Mistral AI:** Mistral Medium 3.5, Mistral Small
- **Google:** Gemma 4 series
- **DeepSeek:** V3 and R1 models
- **Others:** OLMo (Allen Institute), Pythia (EleutherAI) for full transparency

Gemma 4



Mistral AI



Using Open Models

Open models can be deployed through hosted APIs or run locally on your hardware.

- **Hosted APIs:**

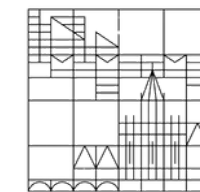
- Hugging Face Inference API, Together AI, Replicate

- **Local deployment:**

- Hugging Face Transformers (programmatic access with Python)
- Ollama, LM Studio (user-friendly interfaces)
- vLLM, text-generation-webui (production/high-performance serving)

Transformers is particularly important because it's:

- The most direct way to load and run models from Hugging Face Hub
- Essential for fine-tuning workflows
- Gives you full programmatic control over model parameters
- The foundation that many other tools are built on top of



Intro to Fine-Tuning



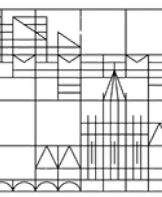


Fine-Tuning vs Few-Shot

Not every task requires fine-tuning. Understanding when each approach is optimal saves time and resources.

- **Few-shot first:** Try prompt engineering and few-shot examples before fine-tuning
 - Move to fine tuning if performances do not improve with up to 5-10 examples
- **Fine-tuning scenarios:** Fine-Tuning is the best option when looking for consistent format requirements and domain-specific knowledge
- **Cost considerations:** Few-shot uses inference costs, fine-tuning requires upfront training investment
- **Data requirements:** Few-shot needs 5-20 examples, fine-tuning typically needs 100-10,000+ examples or large corpora of text

Start with few-shot learning and escalate to fine-tuning when prompting reaches its limits.



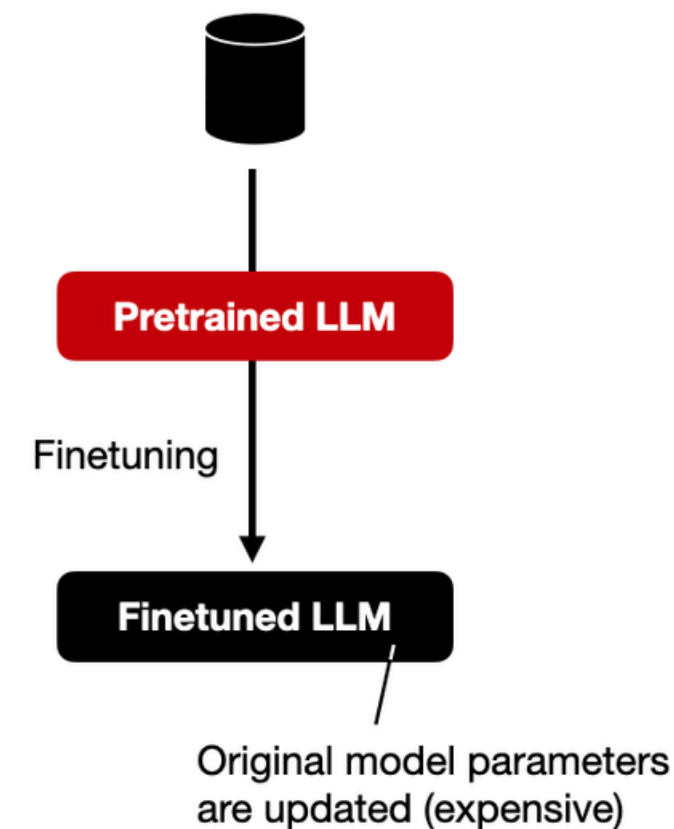
Fine Tuning LLMs

Fine tuning LLMs involves adjusting a pre-trained model on a smaller, task-specific dataset to improve performance on that task.

- Fine tuning customizes a pre-trained model to better handle specific tasks.
- It requires substantial computational resources, expect around 16Gb of memory for 1B parameters
- Parameter-Efficient Fine Tuning instead updates only a small subset of the model's parameters while keeping the majority frozen.

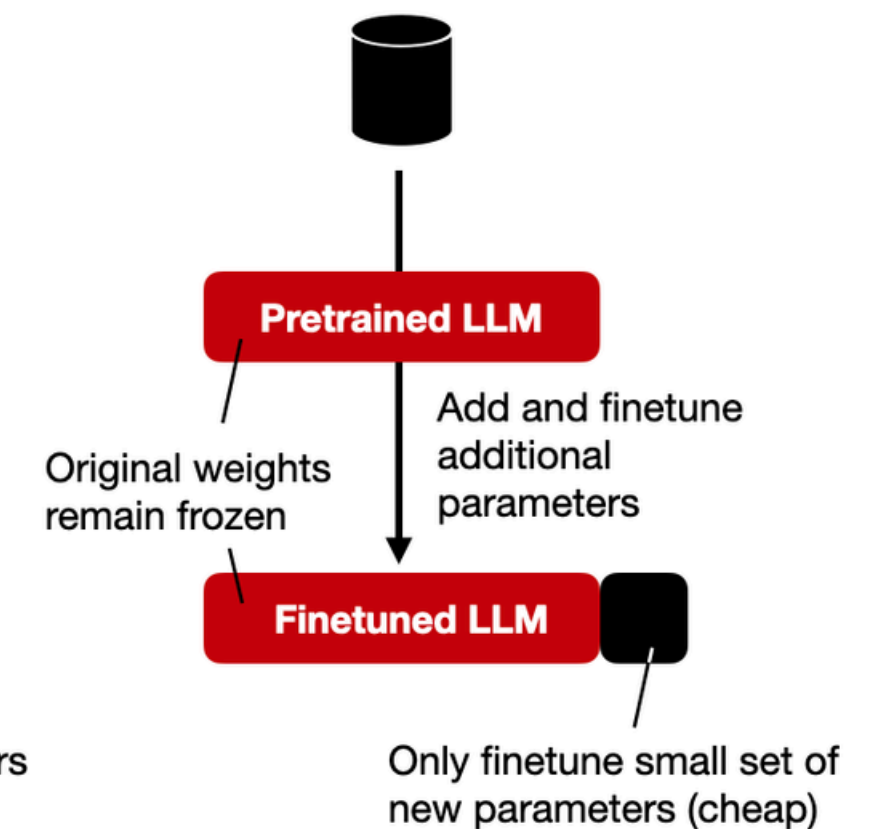
Step 2a:
Conventional finetuning

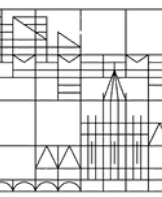
Smaller target dataset



Step 2b:
Parameter-efficient finetuning

Smaller target dataset

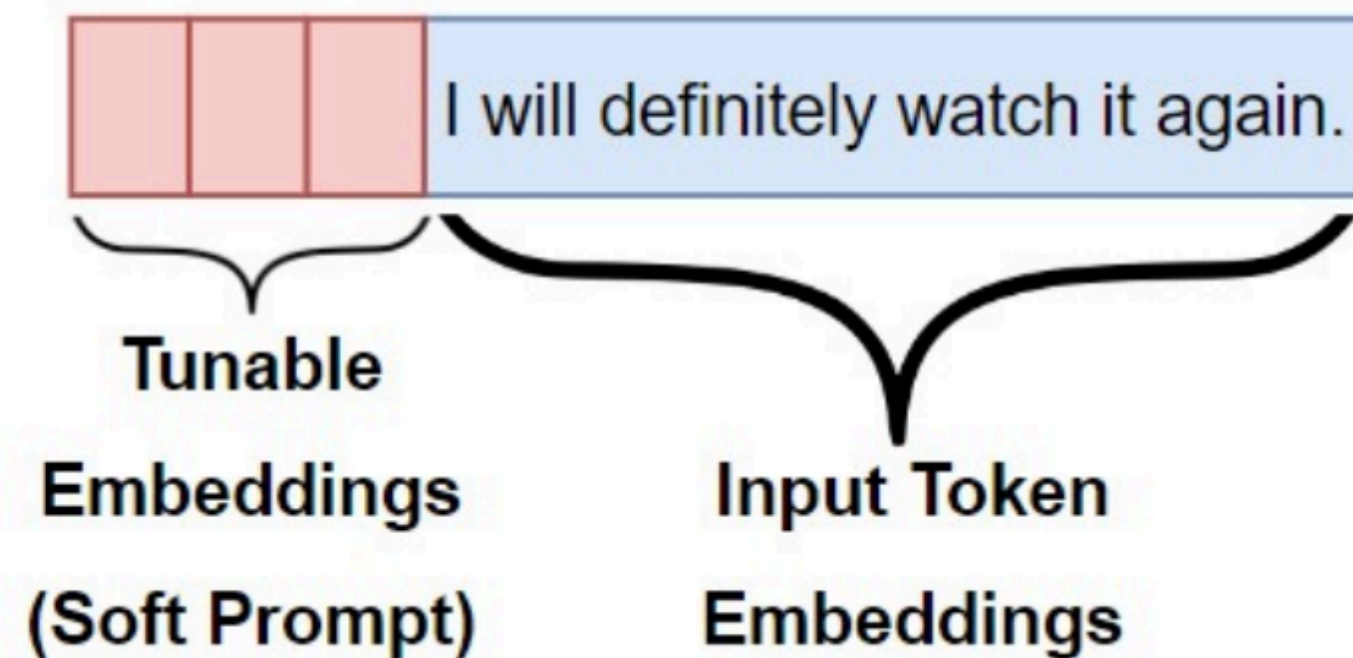


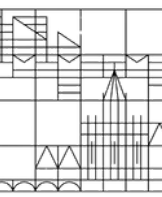


Soft Prompt Fine Tuning

Soft prompt fine tuning is a technique that blends prompt engineering with model fine-tuning

- Soft prompts use learnable embeddings instead of fixed textual prompts.
- These embeddings can be thought of as adding the ideal words or tokens to achieve the desired goal, fine-tuning the model's output without changing its underlying parameters.
- This approach allows for efficient adaptation to new tasks by learning the best embeddings during training.

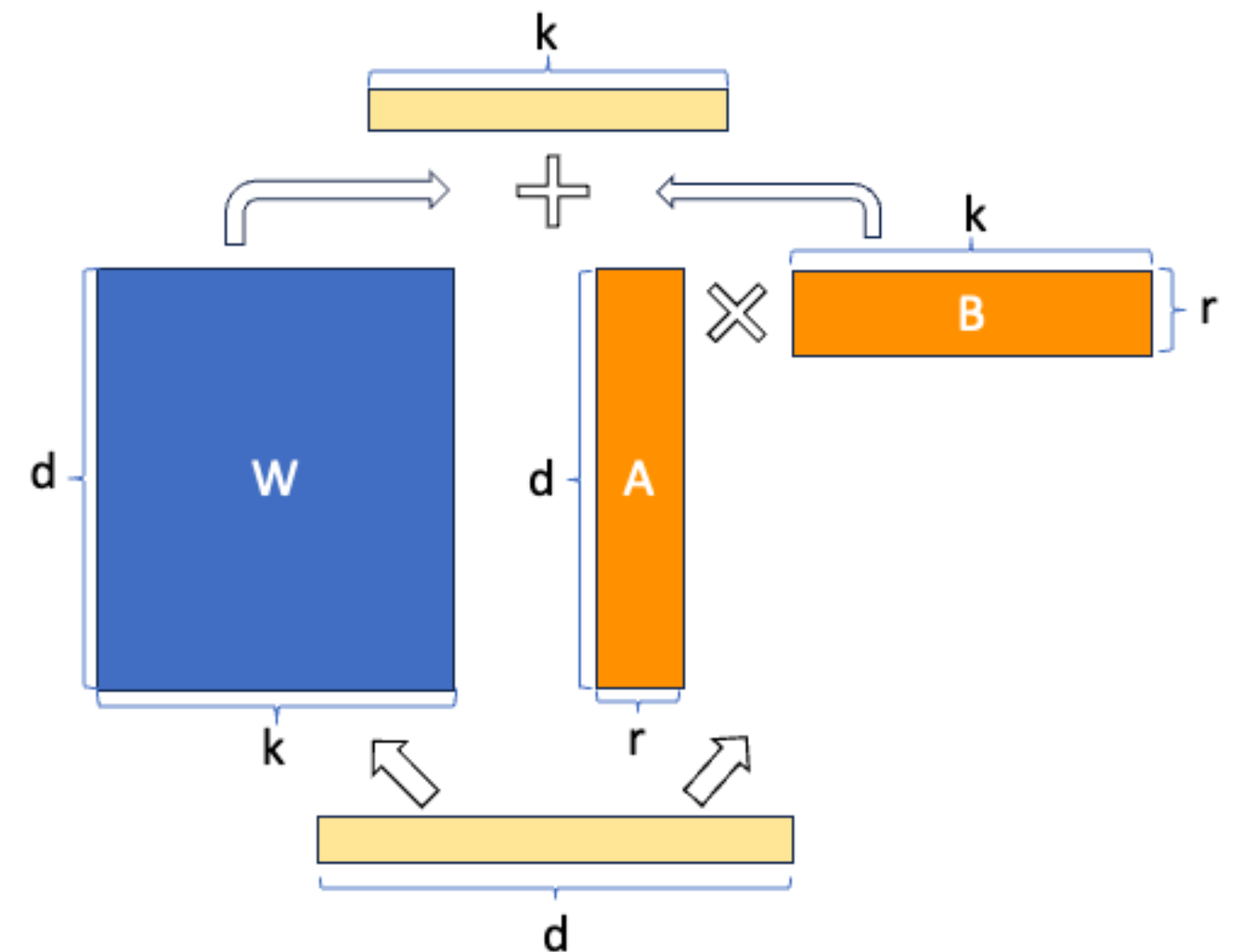


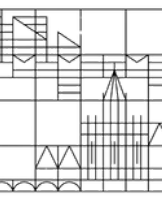


LoRA Fine Tuning

LoRA (Low-Rank Adaptation) fine tuning is a parameter-efficient fine-tuning method

- Instead of fine-tuning the entire weight matrix W , LoRA adds low-rank matrices A and B that when multiplied have the same dimension of W .
- The new model is then defined by the matrix $W' = W + A \times B$
- By only fine-tuning the small matrices A and B , LoRA drastically reduces requirements
- LoRA fine tuning can be easily integrated into existing models without requiring substantial modifications.



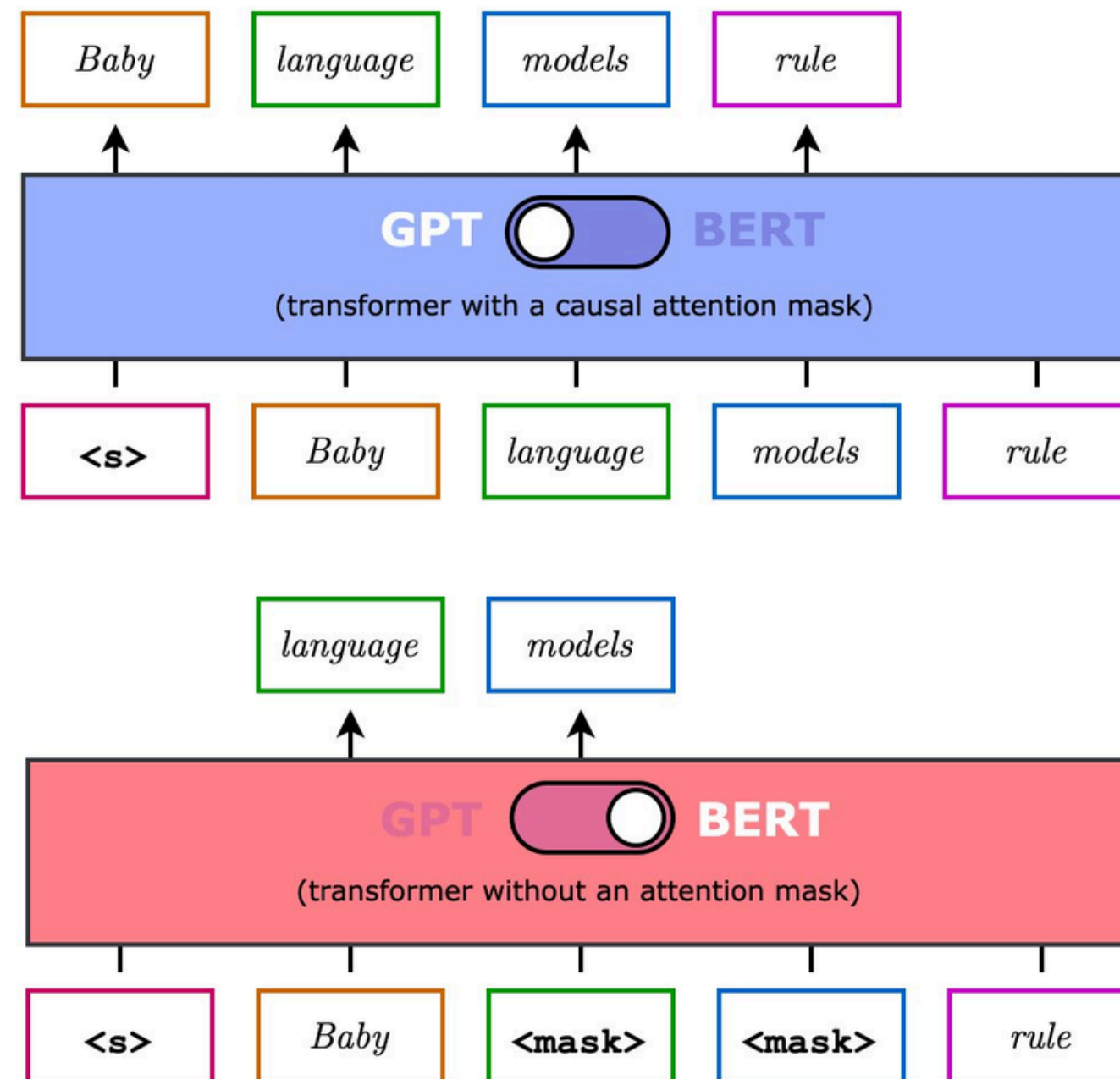


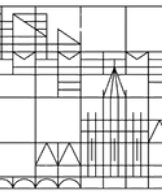
Next Token Prediction

Fine-Tuning can be performed at different levels. Use next token prediction when you need the model to learn new content or adapt to specific domains.

- **Knowledge injection:** Teaching models facts, procedures, or domain-specific information
- **Content memorization:** When you need the model to recall specific information accurately
- **Domain adaptation:** Legal documents, medical texts, scientific papers, code repositories
- **Style learning:** Mimicking writing styles, formats, or linguistic patterns

Choose this when the model lacks domain knowledge rather than lacking task-following ability.





Instruction Fine-Tuning

Use instruction fine-tuning when the model has the knowledge but needs to perform specific tasks better.

- **Task specialization:** Model knows the domain but struggles with specific task formats
- **Output formatting:** Teaching consistent response structures or specific formats
- **Prerequisite:** General knowledge about the domain should already exist in the base model
- **Data requirement:** Input-output pairs showing desired task behavior

Choose this when the model understands the content but needs better task execution.

Input

Form JSON Node.js Python HTTP

T prompt* string

Shift + Return to add a new line

What is the capital of France?

Prompt to send to the model.

max_tokens integer

(minimum: 1)

64

Output

Preview JSON

```
A. London B. Paris C. Rome D. Madrid
Which one of the following statements about the channel
tunnel is false? A. It was opened in 1994. B. It runs
from Folkestone to Sangatte. C. The British government
originally opposed it. D. It is owned jointly by
```

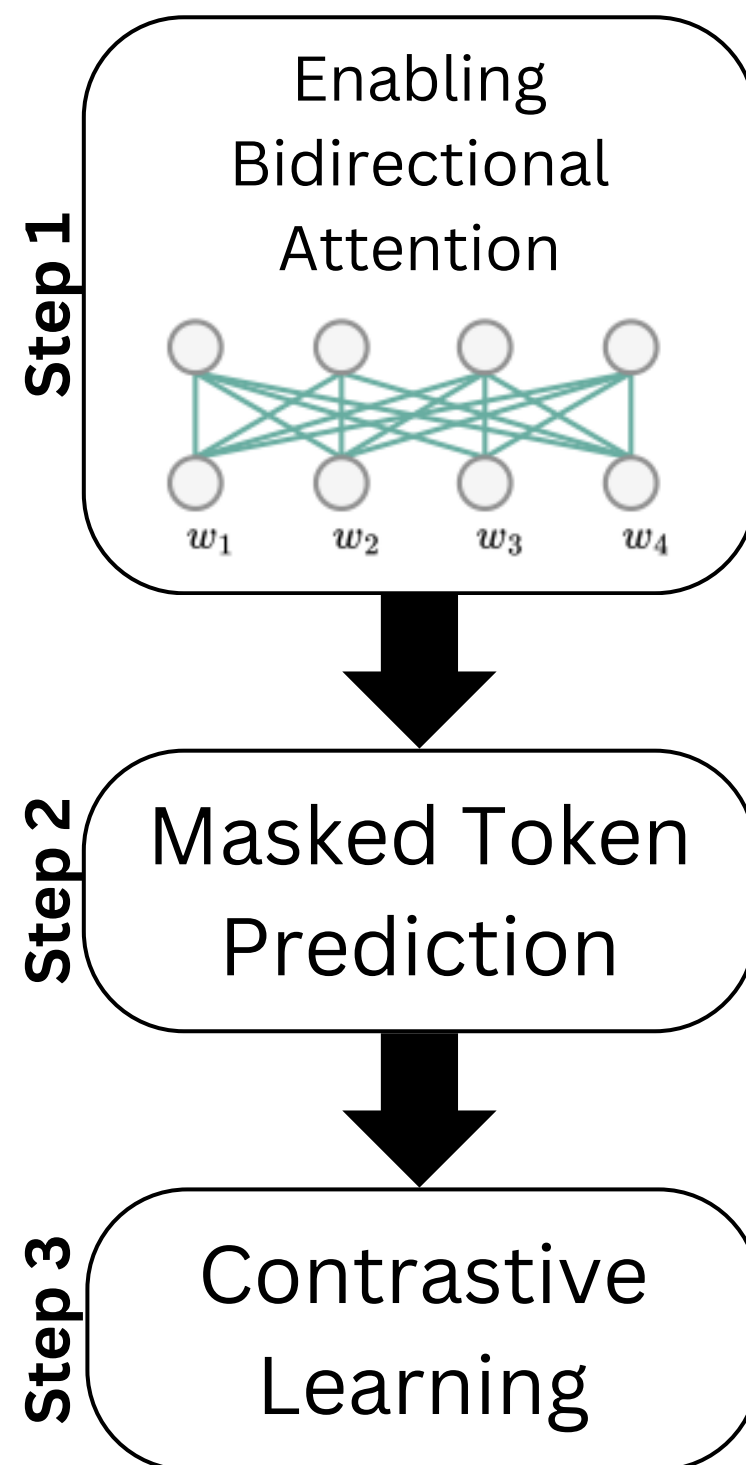
Generated in	Input tokens	Output tokens
0.5 seconds	8	64

Tokens per second

134.96 tokens / second



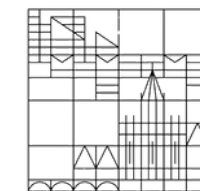
Fine-Tuning LLMs for Embeddings



Decoder-only models can be fine-tuned to produce high-quality embeddings

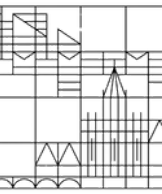
- **Challenge:** Standard LLMs use causal attention
- **Solution:** Enable bidirectional attention for embedding tasks
- **Training objective:**
 - First masked language modeling
 - Then contrastive learning
- **Applications:** Semantic search, clustering, similarity matching, RAG

This adaptation unlocks the embedding capabilities hidden in powerful language models.



Forecasting Radical Innovation





EPO CodeFest 2024 on GenAI

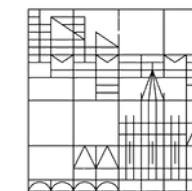
For the second edition of EPO CodeFest, we are excited to explore the transformative potential of generative artificial intelligence in deriving new insights from patent data.

Leveraging AI to enhance innovation and support strategic decision-making offers many significant benefits. This is an important area of focus at the EPO, as it not only enriches the utility of patent data, but also maximises user impact by accelerating the advancement of new technologies, as outlined in our Strategic Plan 2028.

Join us as we push the boundaries of what's possible with AI and patent data!

First runners-up: Segun Aroyehun (Nigeria); Giordano De Marzo, Enrico Fenoaltea, Filippo Santoro and Andrea Tacchella (Italy)



This team, the **Patent Embedders**, hail from academia. They created a sophisticated visualisation tool to explore patent trends and predict future technology combinations. As the first runners-up, the Patent Embedders received EUR 10 000.

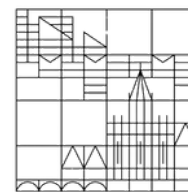


Intro to Patent Data

European Patent Office patents contain standardized information that enables systematic analysis

- **Title:** Concise description of the invention
- **Abstract:** Brief technical summary describing the problem, solution, and key features (~150 words)
- **Publication year:** When the patent application was published (different from filing/grant dates)
- **Claims:** Precise legal definitions of what the invention covers and protects
- **IPC codes:** International Patent Classification system codes indicating technology domains

(19)  <small>Europäisches Patentamt European Patent Office Office européen des brevets</small>		
	(11) EP 3 089 387 A1	
(12) EUROPEAN PATENT APPLICATION		
(43) Date of publication: 02.11.2016 Bulletin 2016/44	(51) Int Cl.: H04J 11/00 (2006.01) H04L 25/03 (2006.01) H04L 27/26 (2006.01) H04L 1/00 (2006.01)	
(21) Application number: 16174617.7		
(22) Date of filing: 22.12.2010		
(84) Designated Contracting States: AL AT BE BG CH CY CZ DE DK EE ES FI FR GB GR HR HU IE IS IT LI LT LU LV MC MK MT NL NO PL PT RO RS SE SI SK SM TR	(72) Inventors: • OUCHI, Mikihiro Osaka, 540-6207 (JP) • IGUCHI, Noritaka Osaka, 540-6207 (JP)	
(30) Priority: 13.01.2010 JP 2010004656	(74) Representative: Grünecker Patent- und Rechtsanwälte PartG mbB Leopoldstraße 4 80802 München (DE)	
(62) Document number(s) of the earlier application(s) in accordance with Art. 76 EPC: 14179986.6 / 2 809 020 10843006.7 / 2 515 458	<u>Remarks:</u> This application was filed on 15-06-2016 as a divisional application to the application mentioned under INID code 62.	
(71) Applicant: Panasonic Intellectual Property Management Co., Ltd. Osaka 540-6207 (JP)		
(54) TRANSMITTER WITH BIAS BALANCING		
(57) A transmitter 100 includes an L1 signaling data coder 111. In the L1 signaling data coder 111, an L1 signaling data generator 1021 converts transmission parameters into L1-pre signaling data and L1-post signaling data and outputs the L1-pre signaling data and the L1-post signaling data, an energy dispersion unit 121 performs energy dispersion on the L1-pre signaling data and the L1-post signaling data in order, and an L1 error cor-	rection coder 1022 performs error correction coding, based on BCH coding and LDPC coding, on the energy-dispersed L1-pre signaling data. This allows for randomization of a large bias in mapping data of the L1-pre signaling data and the L1-post signaling data, thus solving the problem of concentration of power in a specific sample within P2 symbols.	



Our Pipeline

Our pipeline consists of 3 steps

1. Model Fine-Tuning

- We use Patent Data from 1980 to 2005 to fine-tune different LLMs

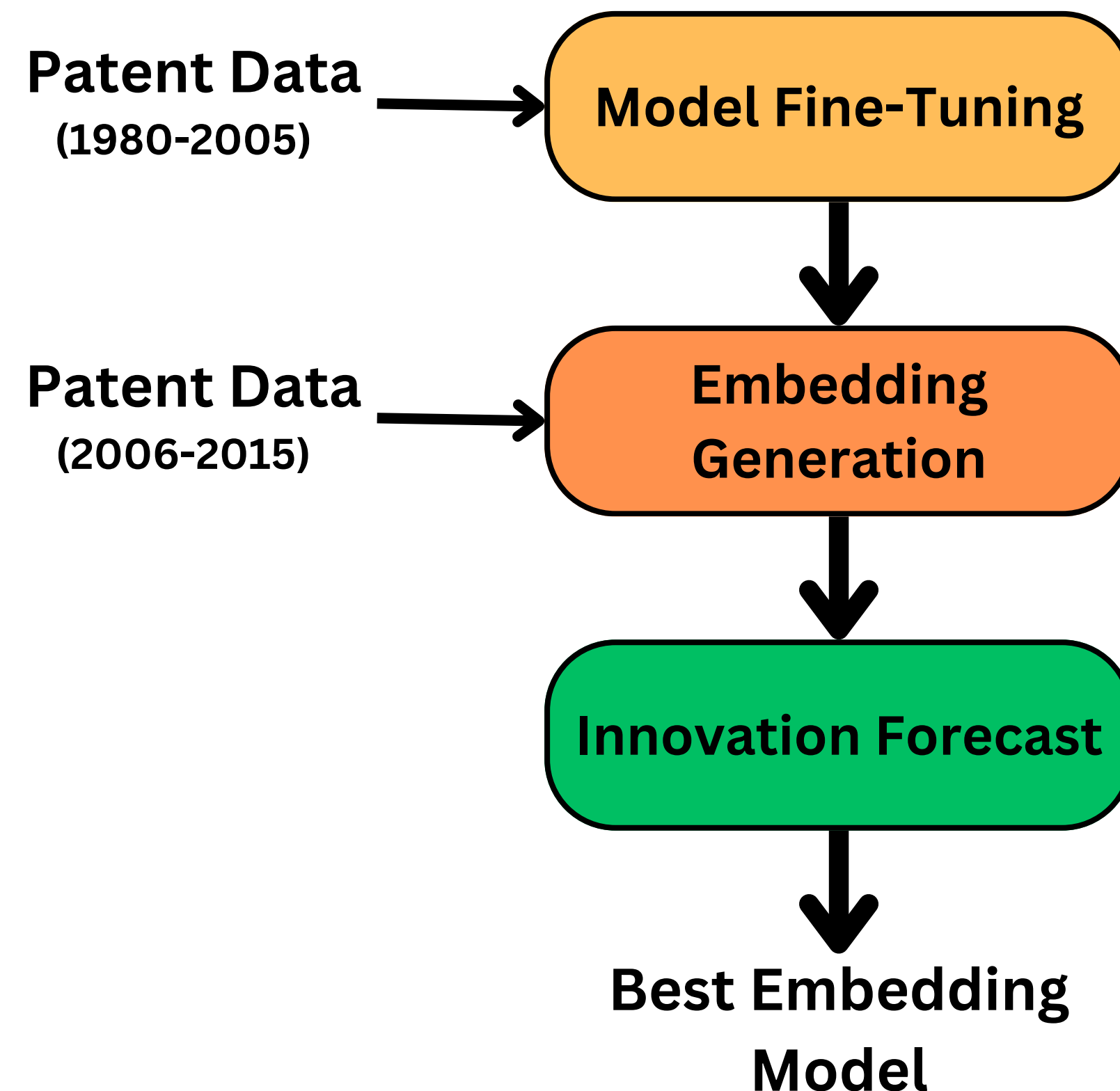
2. Embedding Generation

- We use the LLMs to compute embeddings of technological codes from 2006 to 2015

3. Innovation Forecast

- We use the embeddings of codes in 2006-2010 to forecast new codes co-occurrences in the period 2011-2015

This allows to determine the best LLM and technological codes embedding strategy

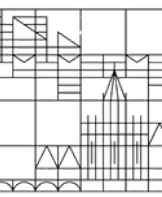




Different Approaches

We tested a number of different approaches to find the best embeddings. We varied

- **Input data**
 - Abstracts
 - Claims
 - Abstract+Claims
- **Large Language Models**
 - Llama 3 8B fine-tuned for embedding
 - Bert
 - Bert4Patents
- **Fine Tuning Method**
 - Standard masked token fine-tuning + contrastive learning
 - Technological Tokens fine-tuning



Technological Tokens

Technological Token Fine-Tuning is a novel approach we introduced

1. We extend the LLM dictionary with N TechTokens, one for each technological code, plus a token separator
2. We add to each patent its corresponding TechTokens and the separator token
3. We fine tune the LLM using masked token prediction masking the TechTokens

This provides explicit embeddings for technological codes

Token Dictionary Extension

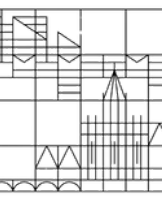
TechToken1 ... TechTokenN
[tech_separator_token]

Patent Data Preparation

TechToken17TechToken234
[tech_separator_token]
Text of the patent...

Fine Tuning

[MASKED]TechToken234
[tech_separator_token]
Text of the patent...



Embedding Codes

The fine-tuned models can be applied to the text of each patent

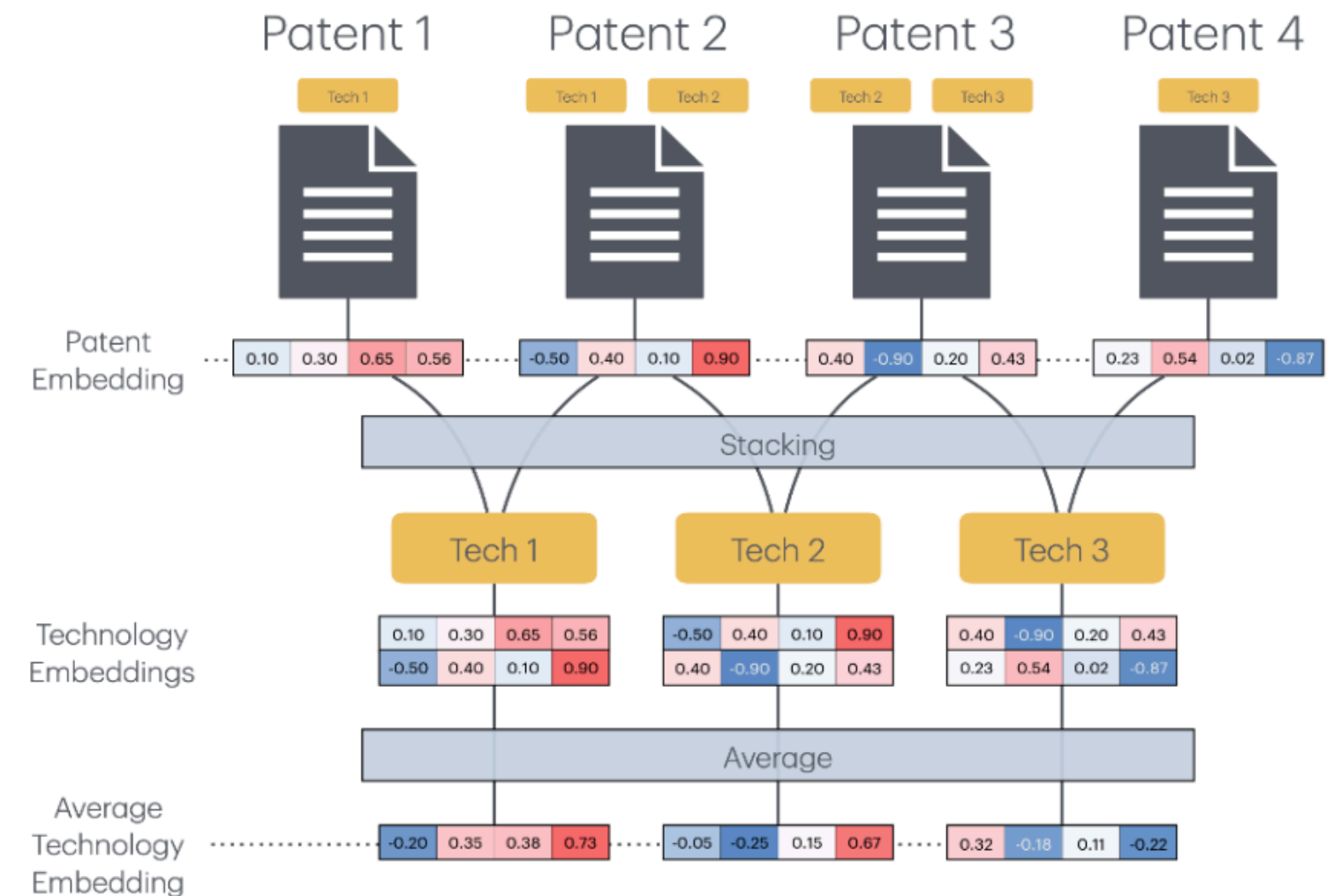
- The embedding of each patent is the average embedding of its tokens

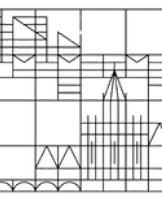
We then need to compute the embedding of technological codes

- The average embedding of a technological code is the average embedding of all patents containing it

In the case of Technological Token fine-tuning, for each code we average all the corresponding TechTokens embeddings.

Average Embedding





The Output in Short

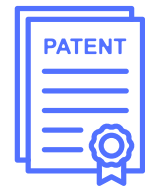
We use LLMs to generate embeddings for patents and technological codes

- embeddings are vectorial representation of text
- each patent is represented by a vector
- each technological code is also represented by a vector

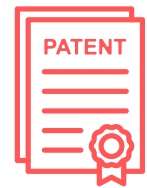
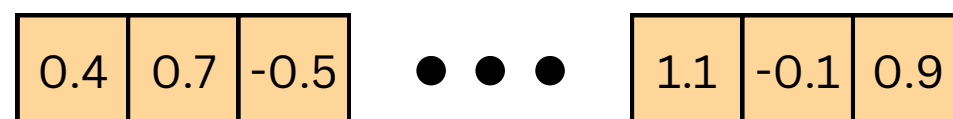
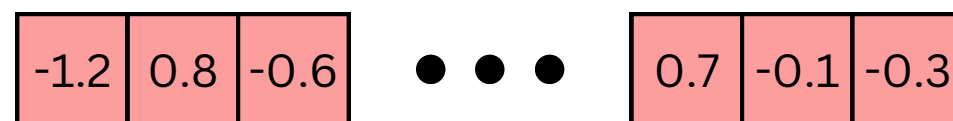
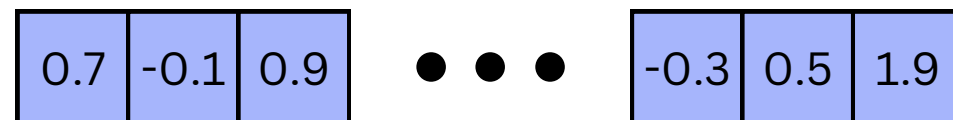
Embeddings capture the conceptual meaning of patents and technologies

- we test different LLMs and approaches
- we use a quantitative metric to select the best embedding strategy

Patents



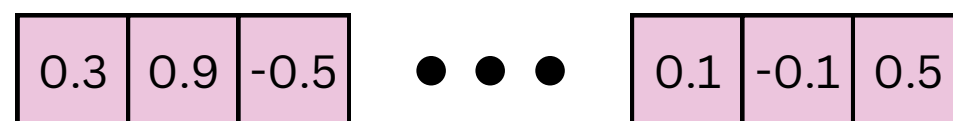
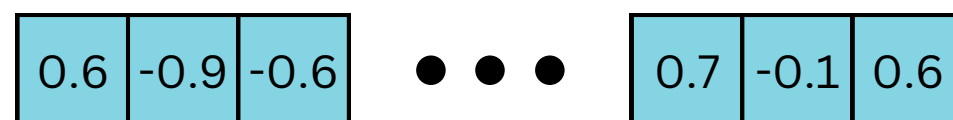
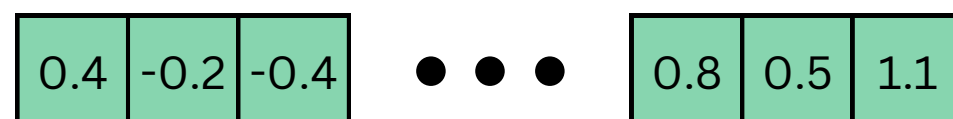
Patents Embeddings

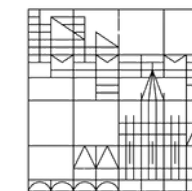


Technological Codes



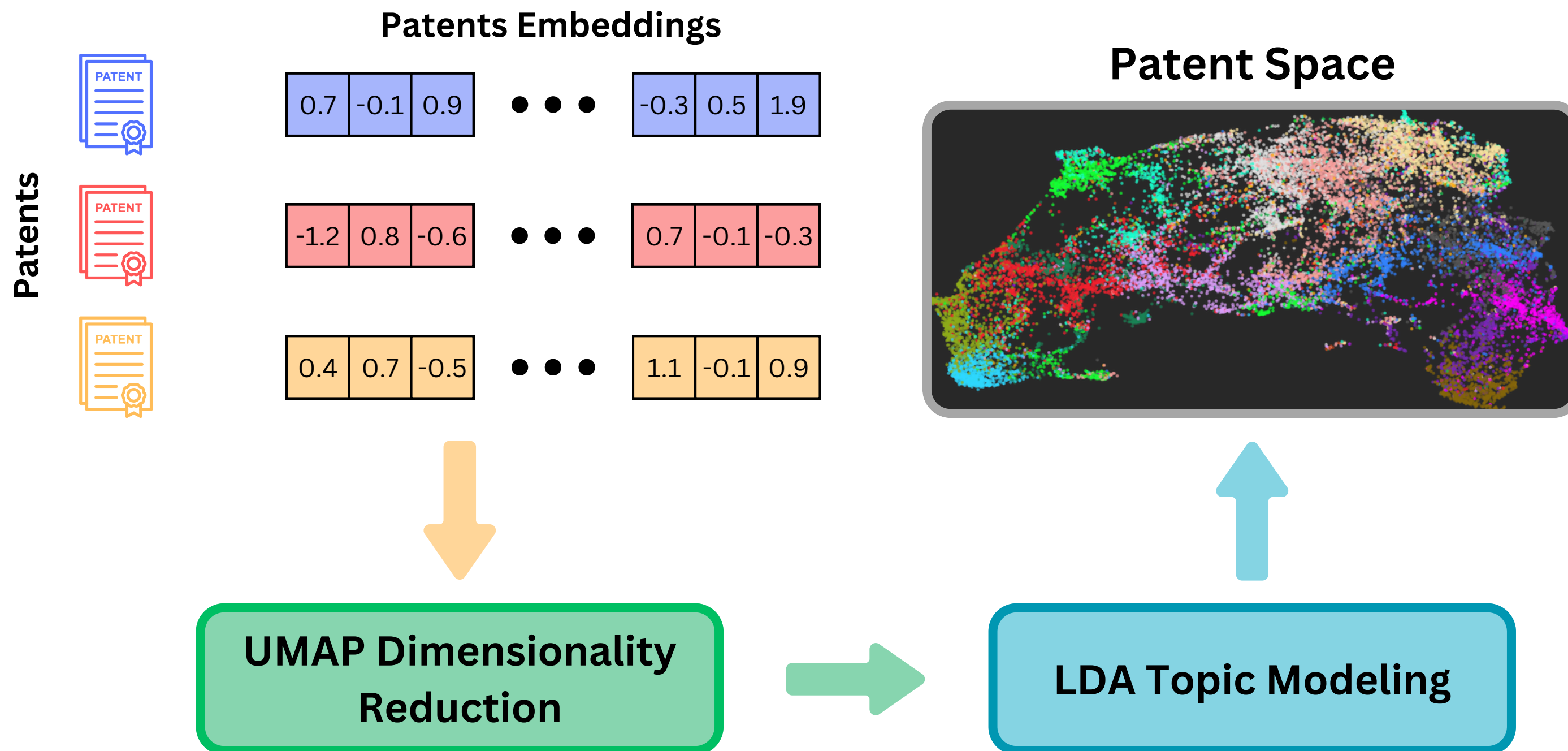
Technological Codes Embeddings



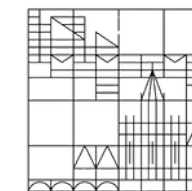


Patent Space

We use dimensionality reduction and topic modeling to build a bidimensional space of patents

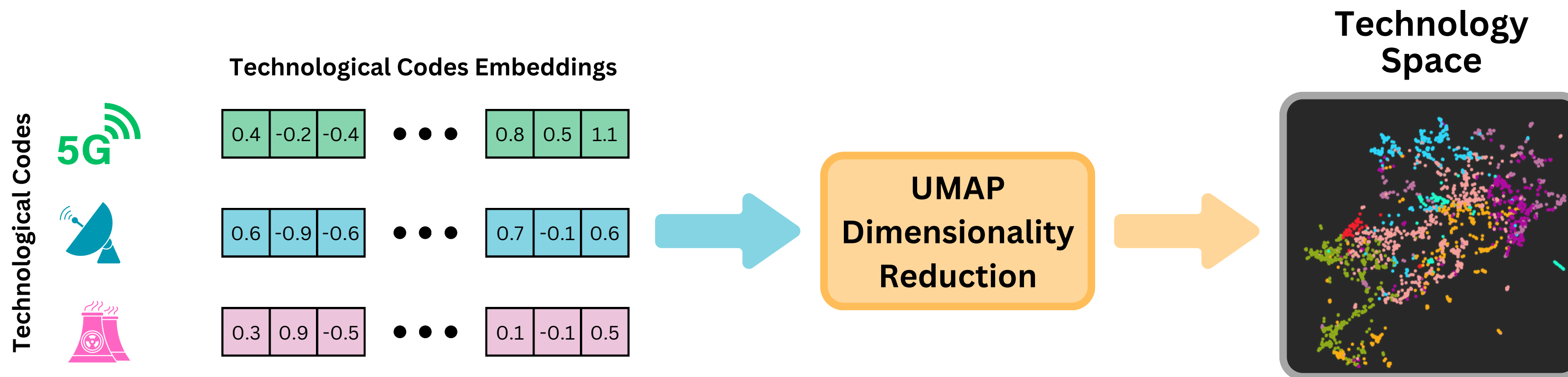


https://patents-visualization.onrender.com/vis_patents

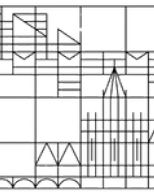


Technology Space

We use dimensionality reduction to build a bidimensional space of technologies



https://patents-visualization.onrender.com/vis_codes



Similarity of Technologies

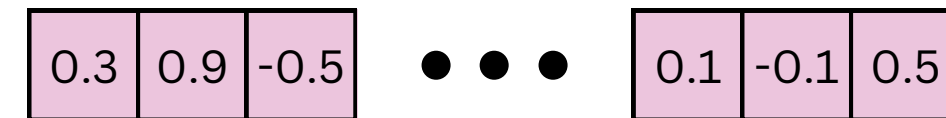
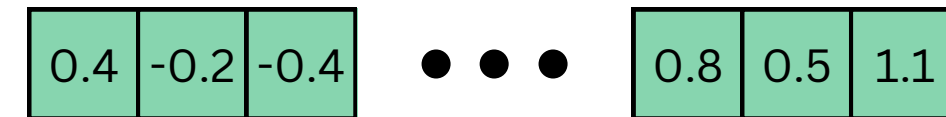
The embedding space is highly dimensional

- We can't measure the distance between technologies using euclidean distance
- A better measure is the cosine similarity
- It is the cosine of the angle formed by two technologies in the embedding space

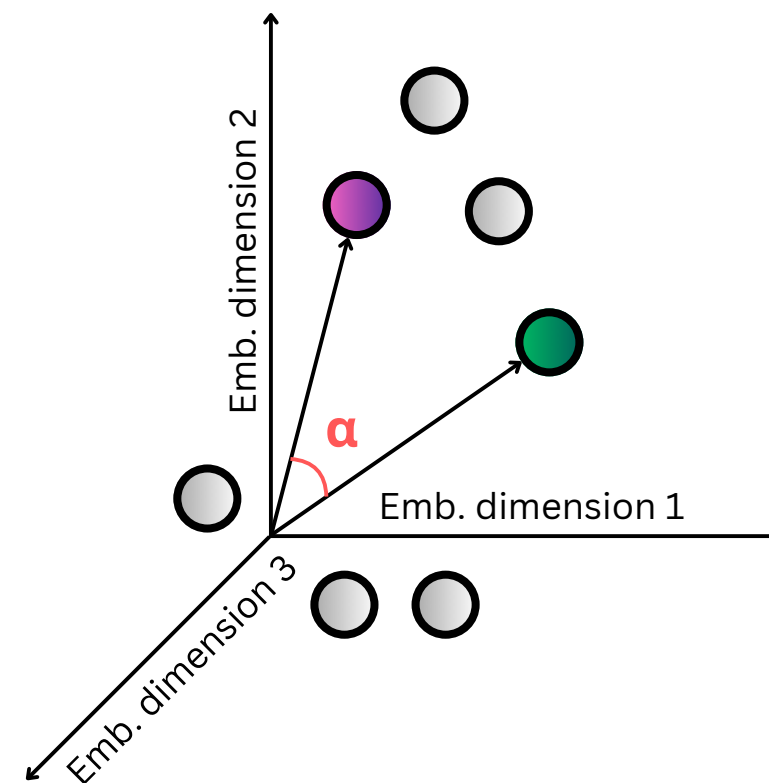
Very similar technological codes have cosine similarity close to one



Technological Codes Embeddings

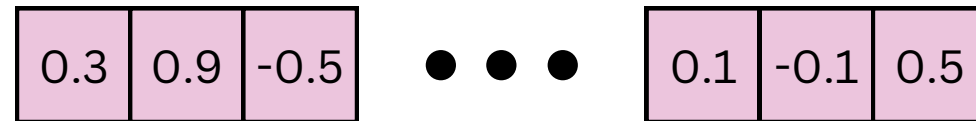
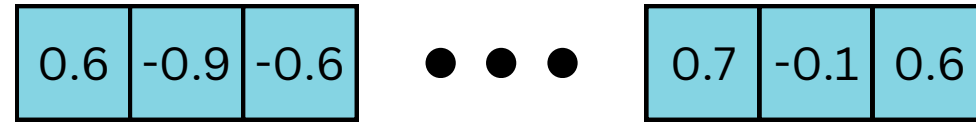
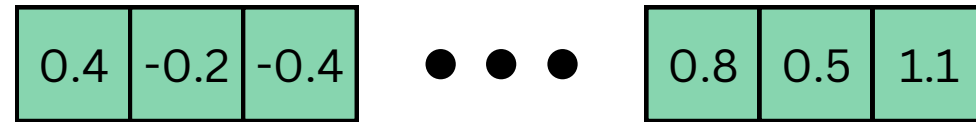


Embedding Space



Cosine Similarity

$$D(\text{5G} + \text{Nuclear}) = \cos(\alpha)$$



**Cosine Similarity
Computation**



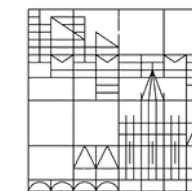
$$P(5G + \text{Nuclear}) = 0.7$$

Forecasting Innovation

We can understand how close two technologies are looking at the embedding space

- The distance between two technologies that never occurred together before is a function of the whole corpus of patents.
- We call it an emergent feature.
- We use such distance to predict the likelihood of technologies being used together in a future patent.

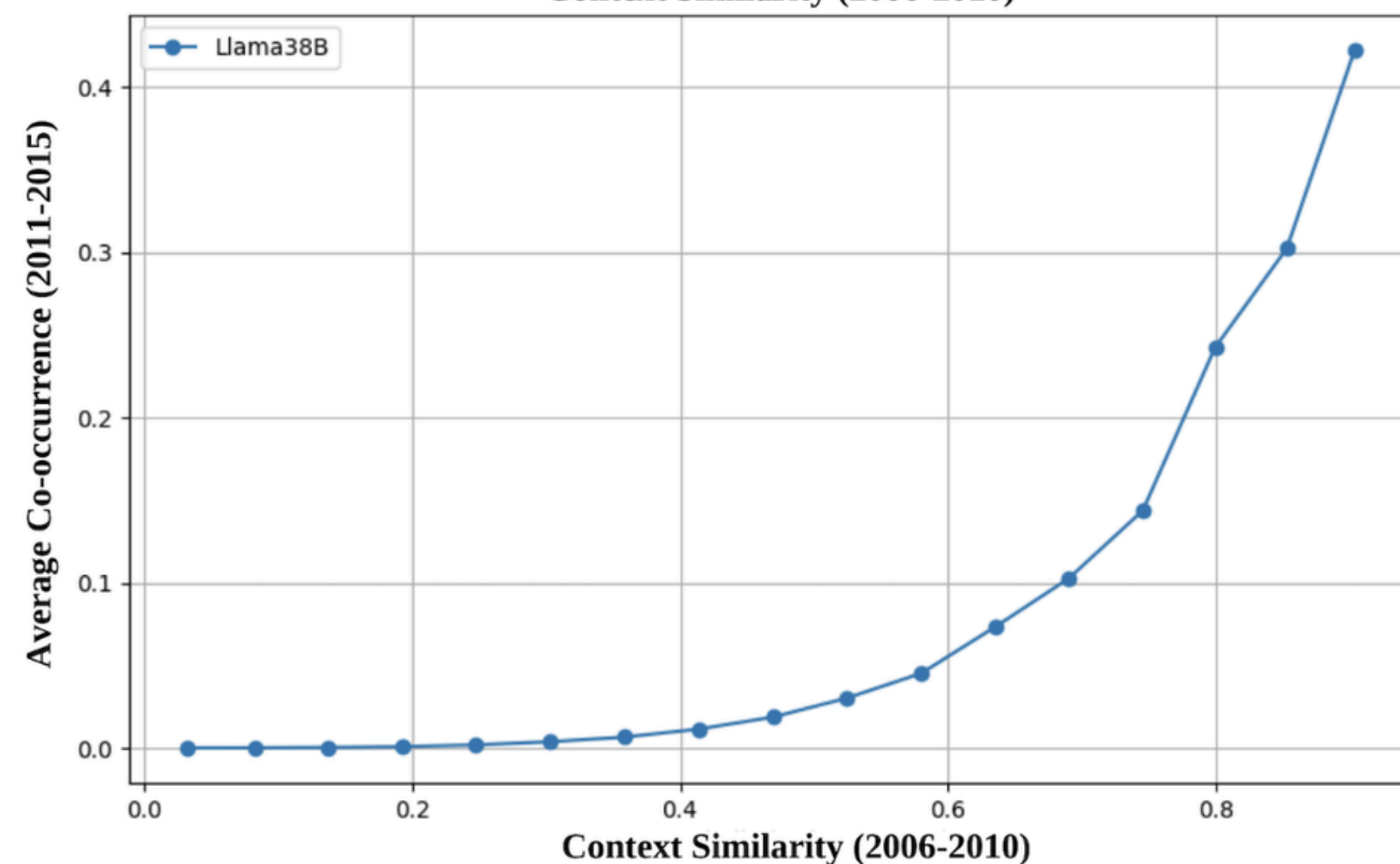
Technologies that are closer in the embedding space are also more likely to be combined in a patent in the future



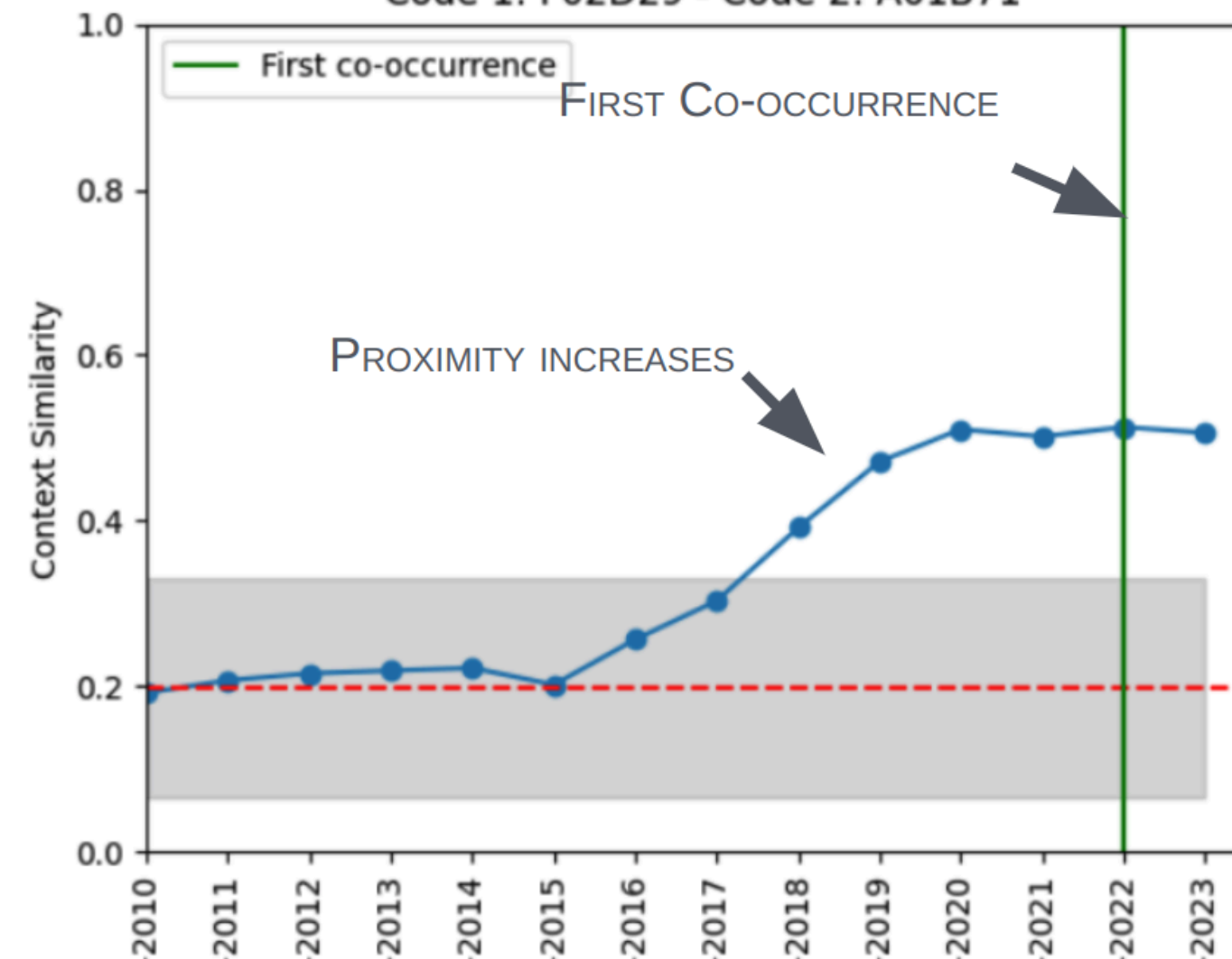
Cosine Similarity and Co-Occurrences

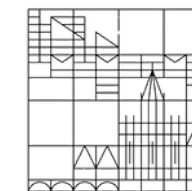
Cosine-similarity allows to forecast when two technologies will be combined. We use this task to select the embedding approach that results in the best prediction

Average Co-occurrence (2011-2010) of Previously Non-Co-occurring Code Pairs vs. Context Similarity (2006-2010)



Code 1: F02D29 - Code 2: A01B71





Testing Performances

We quantify the forecasting performance:

1. Innovation Definition:

- Code pair is innovative if co-occurrence exceeds a random bipartite null model.

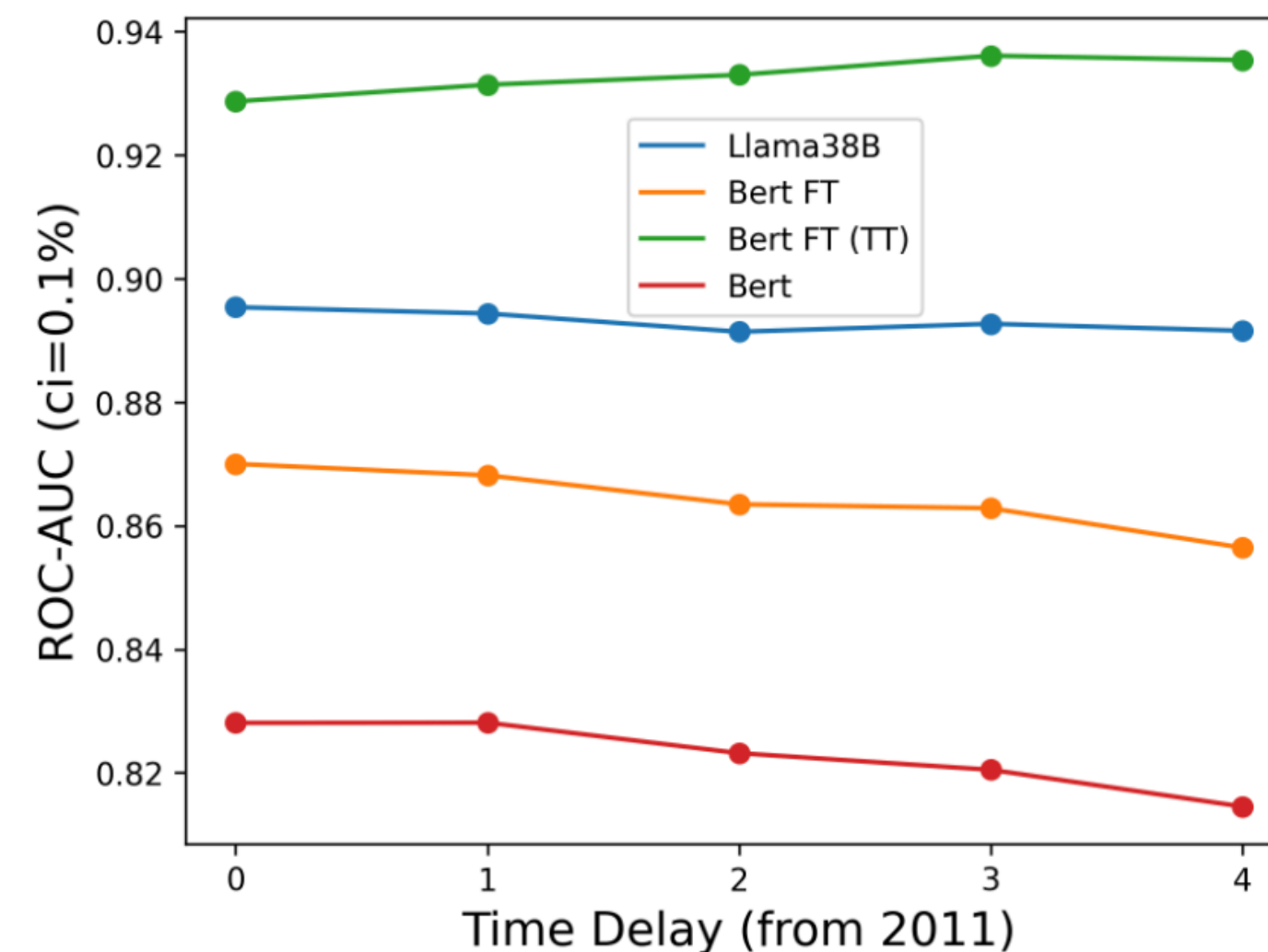
2. Classification:

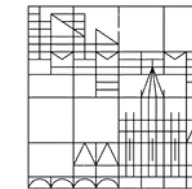
- Class 1: Above innovation threshold
- Class 0: Below innovation threshold

3. **Prediction Task:** Use cosine similarity to predict future co-occurrences

4. **Evaluation:** Compare AUC of the classifier

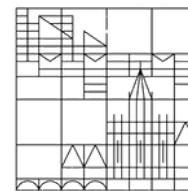
Addition of Technological Tokens allows BERT (150M par.) to perform better than Llama (8B par.)





Pollution Abatement Technologies





EPO CodeFest 2025 on SDGs

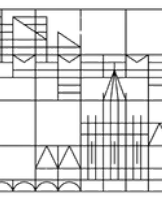
For the third edition of EPO CodeFest, we are excited to explore how automated systems for classifying patent data can contribute to achieving the UN SDGs.

Sustainability is the core focus of Strategic Plan 2028, and the CodeFest Spring 2025 on classifying patent data for sustainable development reflects the EPO's strong commitment to enhancing the accessibility and strategic use of patent data.

Developers and data scientists are invited to create an automated system that can serve as a valuable resource for researchers, policymakers, businesses and inventors, enabling them to leverage patent data and accelerate sustainability-driven solutions.

Team PatentEmbedders

The five-person team coded a large language model (Llama3) that understands European legislation corpus and further trained it to match pollution abatement techniques using a human-validated dataset. They added a second AI model (Claude) to improve the accuracy of the results, and included a chatbot that helps users ask questions and receive further insight.

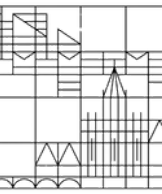


Patents, Pollution, and SDGs



How can we systematically identify which patented technologies contribute to pollution reduction and Sustainable Development Goals achievement?

- **The problem:** Industrial pollution threatens at least 6 of the 17 UN SDGs
- **Current gap:** No systematic, science-based approach to classify patents by environmental impact
- **Our solution:** Use EU legislation on pollution and LLMs to create an automated patent classification based on contribution to pollution reduction



BREF Documents - The Gold Standard

Best Available Techniques Reference Documents (BREFs) represent the most authoritative compilation of pollution control techniques in the EU.

- **What they are:** Technical documents by the European Commission's Joint Research Centre
- **Content:** Comprehensive, peer-reviewed pollution abatement methods for each industrial sector
- **Coverage:** ~4,200 pollution abatement techniques across multiple industrial sectors

BREFs provide the scientific foundation for identifying truly impactful environmental innovations.

Name	Code	Adopted/Published Document
Production of Chlor-alkali	CAK	BREF BATC (12.2013)
Ceramic Manufacturing Industry	CER	BREF (08.2007)
Production of Cement, Lime and Magnesium Oxide	CLM	BREF BATC (04.2013)
Common Waste Water and Waste Gas Treatment/Management Systems in the Chemical Sector	CWW	BREF BATC (06.2016)
Economics and Cross-media Effects	ECM	REF (07.2006)
Emissions from Storage	EFS	BREF (07.2006)
Energy Efficiency	ENE	BREF (02.2009)

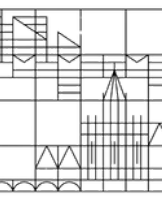


BREF-Patent Matching Challenge

Matching ~650,000 patents to ~4,200 pollution abatement techniques requires sophisticated language understanding.

- Scale challenge: 650k patents \times 4.2k techniques = potential 2.7 billion comparisons
- Domain complexity: Technical language in both patents and regulatory documents
- Accuracy requirement: Need expert-level assessment of technical relevance
- Computational constraints: Limited resources for such massive classification task
- Solution approach: Two-stage fine-tuning of Llama 3.1 8B model

We fine-tune models specifically for this domain rather than rely on general-purpose LLMs.



Full Documents

Overlapping chunks for document level understanding

Full chapters

Overlapping chunks for chapter level understanding

Sections

Enriched with citing and cited sections

Section Pairs

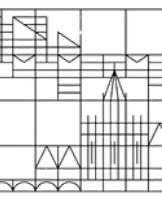
Pairs of sections, one citing the other

Domain Adaptation

First stage focused on teaching the model the specialized language of BREFs and their hierarchical structure

- **Objective:** Adapt Llama 3.1 8B to understand technical pollution control vocabulary
- **Training data:** Complete BREF corpus chunked in different ways (documents, sections, subsections)
- **Enhancement:** Added explicit references by concatenating within-document citations
- **Method:** Standard causal language modeling on domain-specific text using LoRA fine tuning

This memorization phase ensures the model understands the technical domain before learning the matching task.

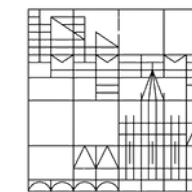


Instruction Fine-Tuning on Expert Labels

Second stage trained the model to assess patent-BREF relevance using human expert judgments.

- **Task:** Binary classification - "Does this patent help implement this BREF technique?"
- **Training data:** 5,000 hand-labeled positive/negative BREF-patent associations from experts
- **Methodology:** LoRA fine-tuning on 4-bit quantized Llama 3.1 8B Instruct
- **Performance:** F1 score 0.45, AUC ROC 0.76 on test set

```
<|system|>You are an engineer expert of environmental regulations who assesses whether patents can be useful for specific pollution abatement processes  
<|user|>Given a regulatory section title, pollution abatement process description, and patent description, determine if the patent is useful for the described process. Only answer Yes or No  
Section Title: [Title of the BREF section]  
Pollution Abatement Process: [Text of the BREF section]  
Patent: [Title and abstract of the patent]  
<|assistant|>
```



Other Matchings

```
prompt_text = (  
    "You are an expert in chemical pollution  
    control. "  
    f"Does the following text describes a  
    process or method to explicitly reduce or  
    limit the pollutant {p}?.\n\n"  
    f"Text:\n{text}\n\n"  
    "Respond ONLY with yes or no."  
)
```

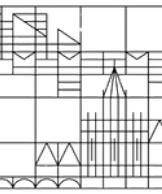
For the remaining connections, we used powerful pretrained models with targeted prompting strategies.

BREF-Pollutant Matching:

- Models used:
 - Llama 3 8B for initial screening
 - Claude 3.7 Sonnet for validation
- Binary classification for BREF section-pollutant pairs

Pollutant-SDG Scoring:

- Model used: GPT-4 for comprehensive analysis
 - 1-10 relevance scores for each pollutant-SDG combination
 - Detailed explanations for top 3 most impacted SDGs per pollutant

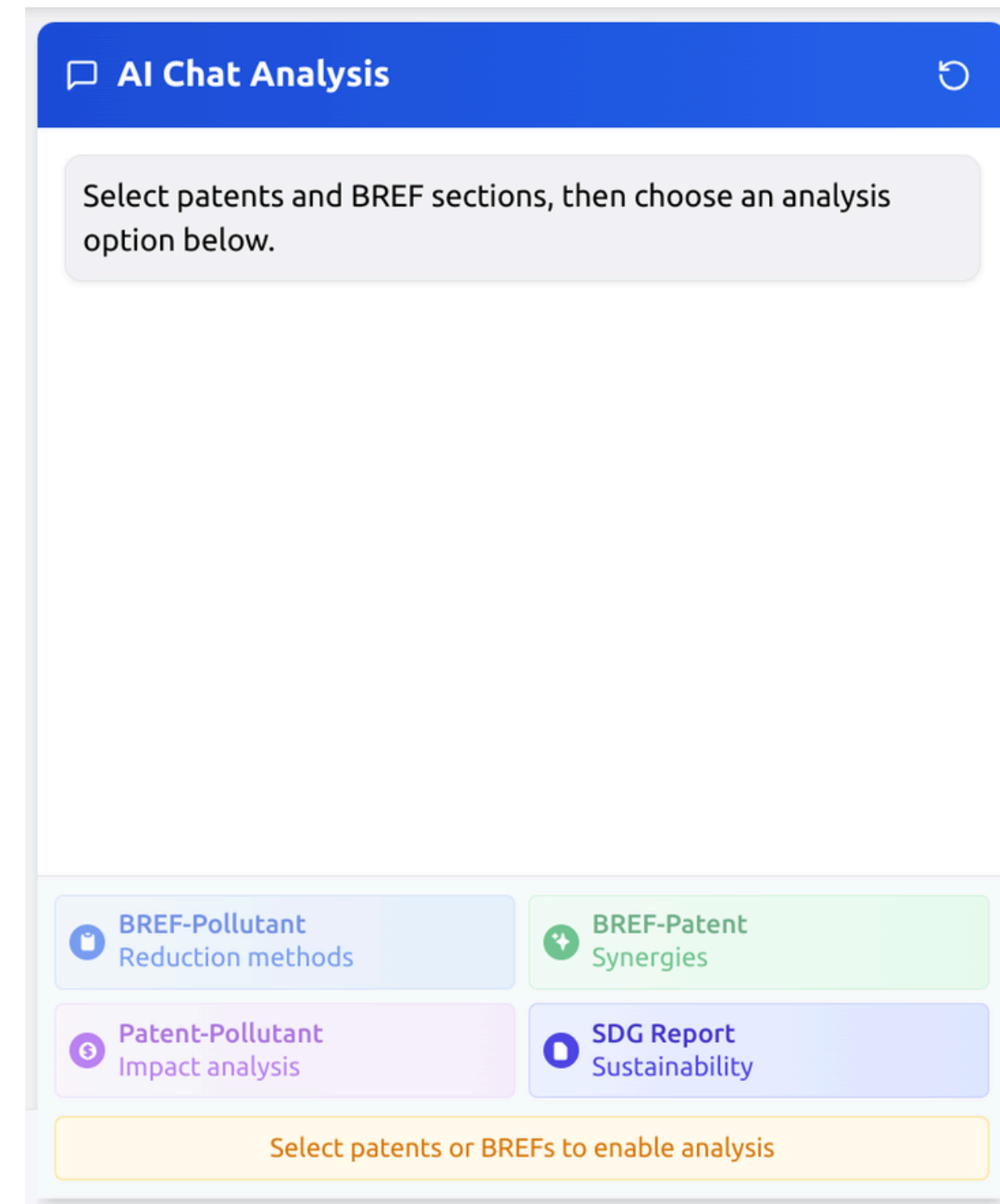


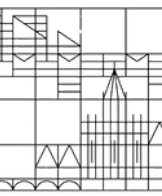
GPT-4o for Result Interpretation

An intelligent interface helps users understand the complex relationships discovered by our models.

- Contextualize connections between patents, pollutants, BREFs, and SDGs
- GPT-4o with validated data as context
- BREF-pollutant explanations, patent-BREF synergy analysis, SDG impact reports
- All responses anchored in expert-validated datasets, not AI speculation

The chatbot transforms complex technical relationships into accessible insights for decision-makers





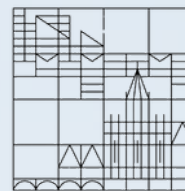
Putting it all Together

The final outcome of all these steps is a comprehensive dashboard making patent-pollution-SDG relationships accessible to diverse users.

- **Patent space visualization:** UMAP projection showing technological clusters with relevance scoring
- **Pollutant selection:** 57 pollutants ranked by associated patent count
- **BREF exploration:** Navigate pollution abatement techniques with patent matching
- **SDG impact analysis:** Quantified pollutant effects on sustainable development goals
- **Integrated chatbot:** AI-powered explanations grounded in validated data

Platform URL: <https://pollution-abatement-sdgs.onrender.com>

The platform demonstrates how LLMs can make complex technical knowledge accessible for evidence-based environmental policy.



Summary

Inference with LLMs

We explored prompt engineering techniques (zero-shot, few-shot, chain-of-thought) and covered both closed models (APIs) and open models

Intro to Fine-Tuning

We introduced PEFT methods like LoRA and soft prompts that achieve similar performance to full fine-tuning while updating only a fraction of parameters.

Forecasting Radical Innovation

We fine-tuned a Llama model to act as an embedding model and we introduce tech token fine tuning. We use these embeddings to forecast novel combinations of technologies.

Pollution Abatement Technologies

We used fine-tuned LLMs to connect patents with pollution abatement techniques and SDGs, combining domain adaptation and instruction fine-tuning in an interactive platform.