

# Retrieval Augmented Generation in PHP

Michał Żarnecki



## Technical leader at CompanyHouse AG

- Big data / Data mining / NLP / ML
- Python / PHP
- Real-time processing of trade register data
- Credit reports, owner information
- Enterprise structure



Michał Żarnecki



UCZELNIA LUDZI CIĘKAWYCH

## Lecturer, Department of Computer Science and Data Analysis Faculty of Applied Sciences

- Modules:  
"Machine learning in Python", "Text data mining", "Gen AI with LLMs"
- E-learning course:  
*"Machine Learning - How to use the potential of data,  
to get better results and make smarter decisions"*

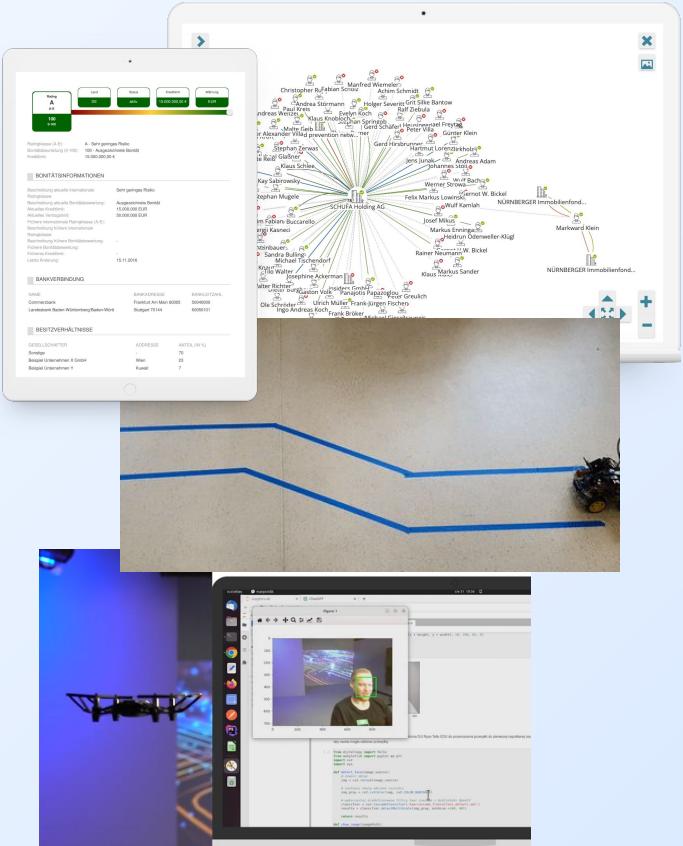


2

<https://github.com/mzarnecki>

Medium

<https://medium.com/@brightcode>



# Resources

Me Article on medium.com



<https://shorturl.at/FmlJt>

 Github repository



<https://github.com/mzarnecki/php-rag>

 Claude  
3.5 Sonnet

 OpenAI  
ChatGPT 4.0

 Gemini

 Llama 3.3

 MISTRAL  
AI\_

 deepseek

# What I can build using RAG?

 Codebase expert & reviewer

 Customer Support Expert

 Content Fact-Checker

 Research Paper Assistant

 Personalized Learning Companion

 Legal Case Assistant

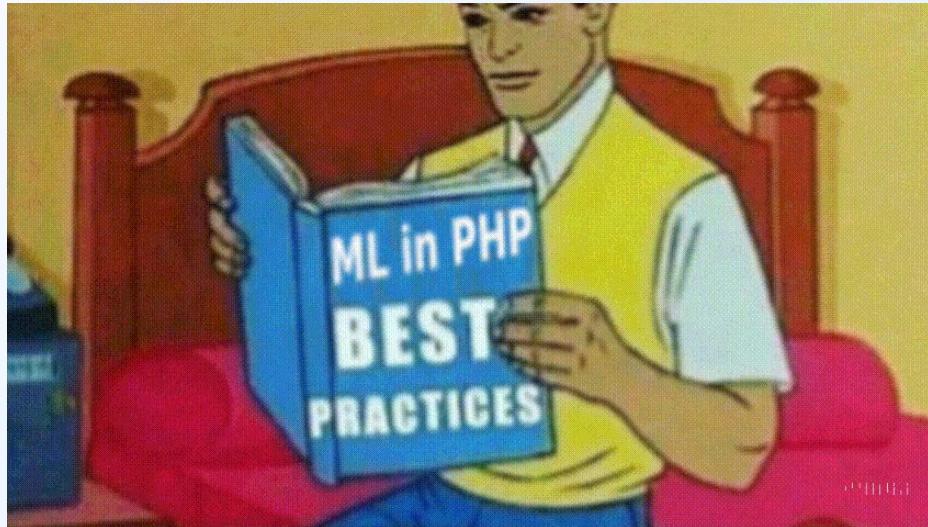
 Manufacturing Troubleshooter

 Travel Itinerary Planner

 Documentation Agent

 Medical Literature Assistant

# Is it worth to use Machine Learning in PHP?



animated with SORA

# ML libs in PHP vs Python

Python

	Library	GPU / CUDA / TPU	Contributors	Year of creation	Github
General-Purpose Machine Learning	Scikit-learn	No	2971	2007	<a href="#">link</a>
	H2O	GPU / CUDA	20	2015	<a href="#">link</a>
	pycaret	GPU / CUDA	126	2020	<a href="#">link</a>
Deep Learning	PyTorch	GPU / CUDA / TPU	3629	2016	<a href="#">link</a>
	TensorFlow	GPU / CUDA / TPU	3637	2015	<a href="#">link</a>
	Keras	GPU / CUDA / TPU	1284	2015	<a href="#">link</a>
Gradient Boosting	xgboost	GPU / CUDA	627	2014	<a href="#">link</a>
	lightgbm	GPU / CUDA	321	2016	<a href="#">link</a>
	CatBoost	GPU / CUDA	377	2017	<a href="#">link</a>
Natural Language Processing (NLP)	SpaCy	GPU / CUDA	681	2015	<a href="#">link</a>
	NLTK	No	397	2001	<a href="#">link</a>
	Gensim	No	431	2009	<a href="#">link</a>
Optical Character Recognition (OCR)	Tesseract	No	45	2005	<a href="#">link</a>
	OCRopus	No	27	2007	<a href="#">link</a>
	OpenCV	GPU / CUDA	36	2000	<a href="#">link</a>
Transformers	Transformers	GPU / CUDA / TPU	2888	2017	<a href="#">link</a>

PHP

Library	GPU/CUDA/TPU	Contributors	Algorithms	Year of creation	Github
RubixML	No	40	40+	2018	<a href="#">link</a>
PHP-ML	No	32	10+	2016	<a href="#">link</a>
FANN	GPU	30	NN*	2003	<a href="#">link</a>
DeepDetect	GPU	26	30+	2015	<a href="#">link</a>
Rindow	GPU**	1	NN*	2013 (2020)	<a href="#">link</a>

\* Focused on Neural Networks

\*\* GPU acceleration using OpenCL

Language	No. of supported algorithms
PHP	dozens
Python	all known

# LLM/RAG libs in PHP vs Python

Python

Library	Contributors	Models	Year of creation	Github
LangChain	3375	all known	2023	<a href="#">Link</a>
Llmalndex	1373	all known	2023	<a href="#">Link</a>
Transformers	2888	all huggingface	2017	<a href="#">Link</a>
Diffusers	878	diffusion models	2022	<a href="#">Link</a>
Ilmware	73	SLIM, BLING, DRAGON, Industry-BERT	2023	<a href="#">Link</a>

PHP

Library/project	Contributors	Models	Year of creation	Github
LLPhant	37	Mixtral, Llama, GPT-4o, Claude	2024	<a href="#">Link</a>
OpenAI PHP	54	OpenAI models	2024	<a href="#">Link</a>
gemini-api-php	-	Gemini family	2023	<a href="#">Link</a>
mzarnecki/php-rag	2	GPT-4o, Gemini 2.0 Flash, Claude 3.5 Sonnet, Llama 3.3, Mixtral, Bielik	2024	<a href="#">Link</a>
Krisseck/php-rag	1	OpenAI, Replicate, KoboldAI Horde	2023	<a href="#">Link</a>

*In the end model is stored as binary data accessed via API and high performance computations are in C....*

## Technical leader at CompanyHouse AG

- Big data / Data mining / NLP / ML
- Python / PHP
- Real-time processing of trade register data
- Credit reports, owner information
- Enterprise structure



UCZELNIA LUDZI CIĘKAWYCH

## Lecturer, Department of Computer Science and Data Analysis Faculty of Applied Sciences

- Modules:  
"Machine learning in Python", "Text data mining", "Gen AI with ChatGPT", "MLMs"
- E-learning course:  
"Machine Learning - How to use the potential of data,  
to get better results and make smarter decisions"



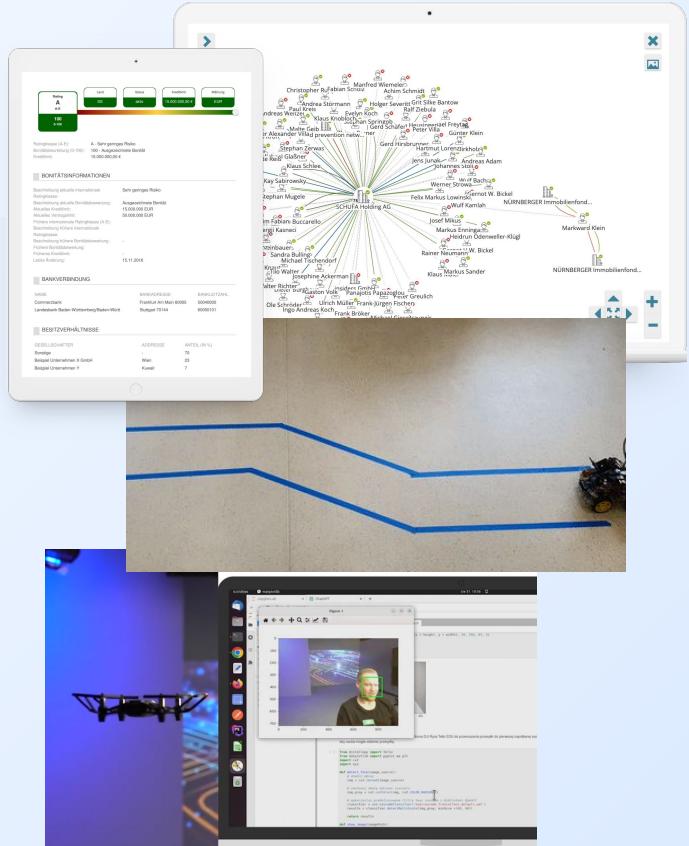
GitHub

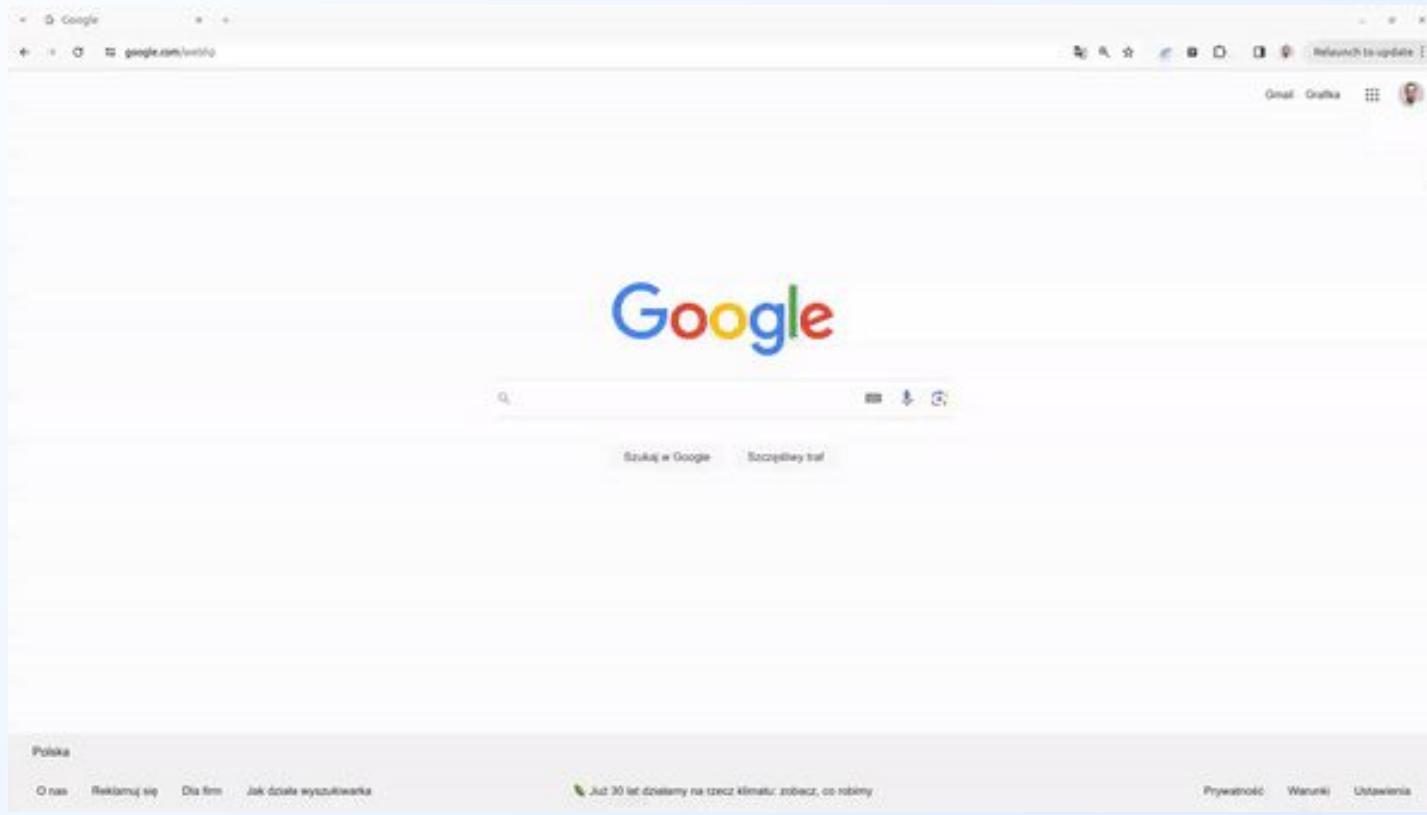
8

<https://github.com/mzarnecki>

Medium

<https://medium.com/@brightcode>







Michał Żarnecki

Audio engineer : [View profile](#)

Michał Żarnecki (ur. 12 listopada 1946 w Warszawie, zm. 21 listopada 2016, tamże) – **polski operator i reżyser dźwięku.**

[Wikipedia](#)  
[https://pl.wikipedia.org › wiki › Michał\\_Żarnecki](https://pl.wikipedia.org/w/index.php?title=Michał_Żarnecki&oldid=11000000)

[Michał Żarnecki – Wikipedia, wolna encyklopedia](#)

**Born:** November 12, 1946, Warsaw

**Died:** November 21, 2016, Warsaw

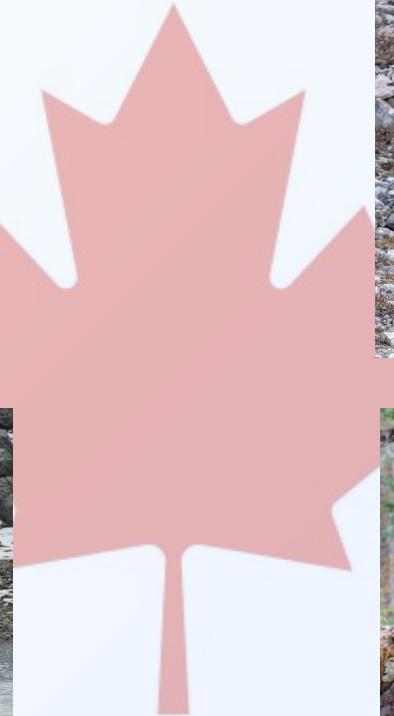
**Nominations:** Polish Academy Award for Best Sound

**Siblings:** Andrzej Zarnecki

# How to avoid such mistakes?

# How to avoid such mistakes?

Use model that understands context  
supply knowledge with **transfer learning**



# Question

**mouse-deer?**  
**(chevrotain)**



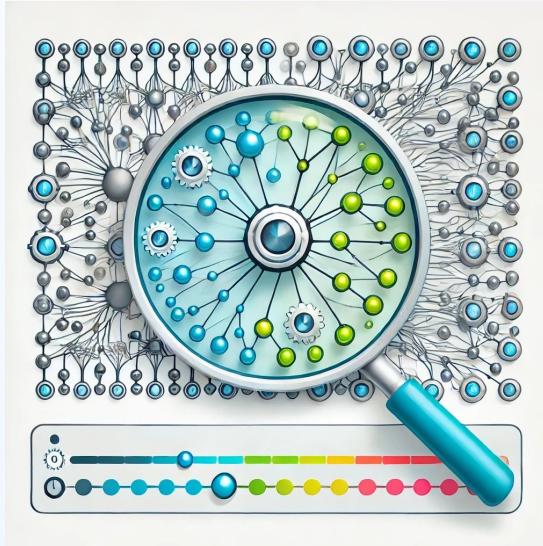




# Support LLM with guideline texts

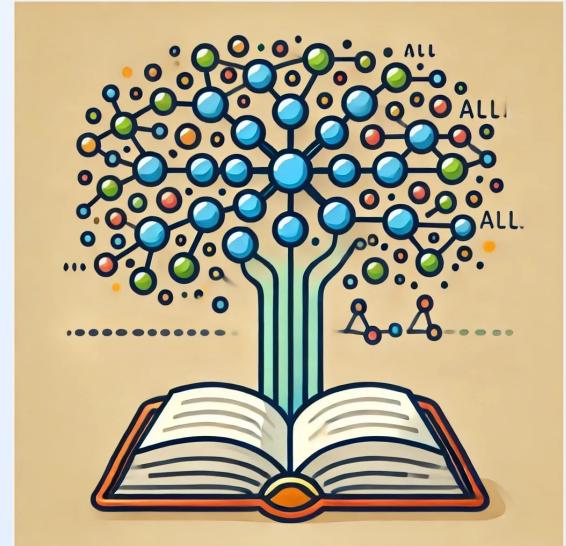
## PEFT or RAG

PARAMETER EFFICIENT FINE TUNING



generated with DALL-E

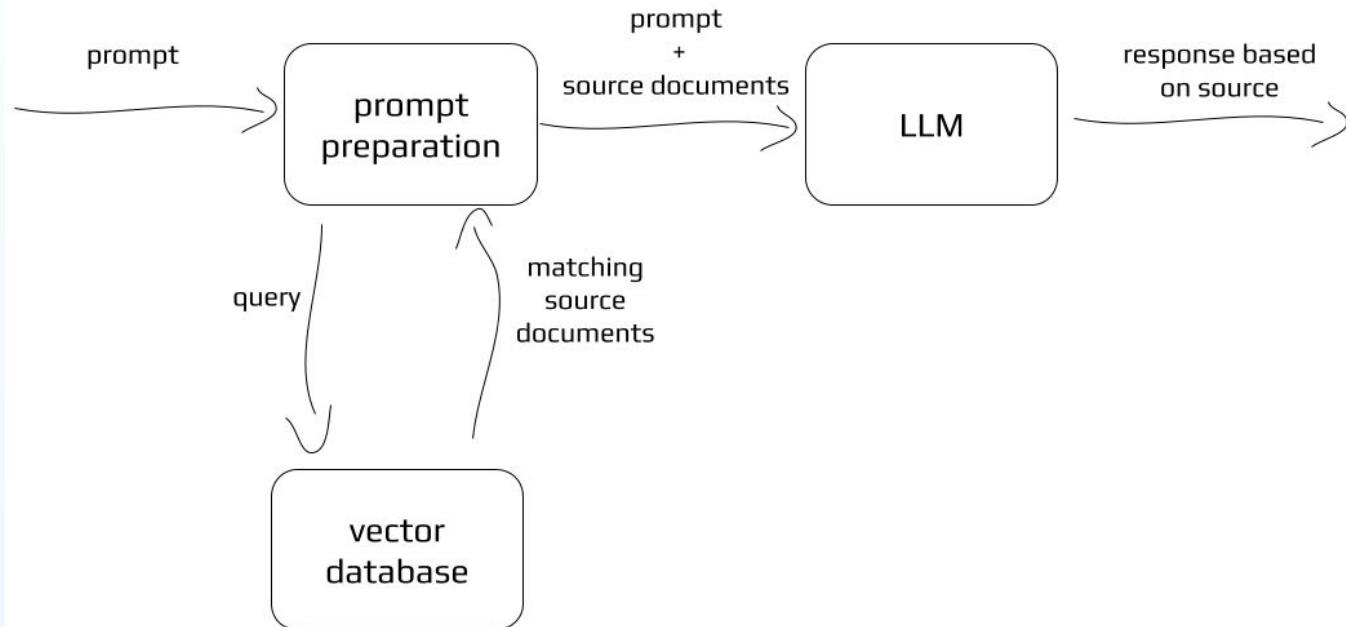
RETRIEVAL AUGMENTED GENERATION



generated with DALL-E

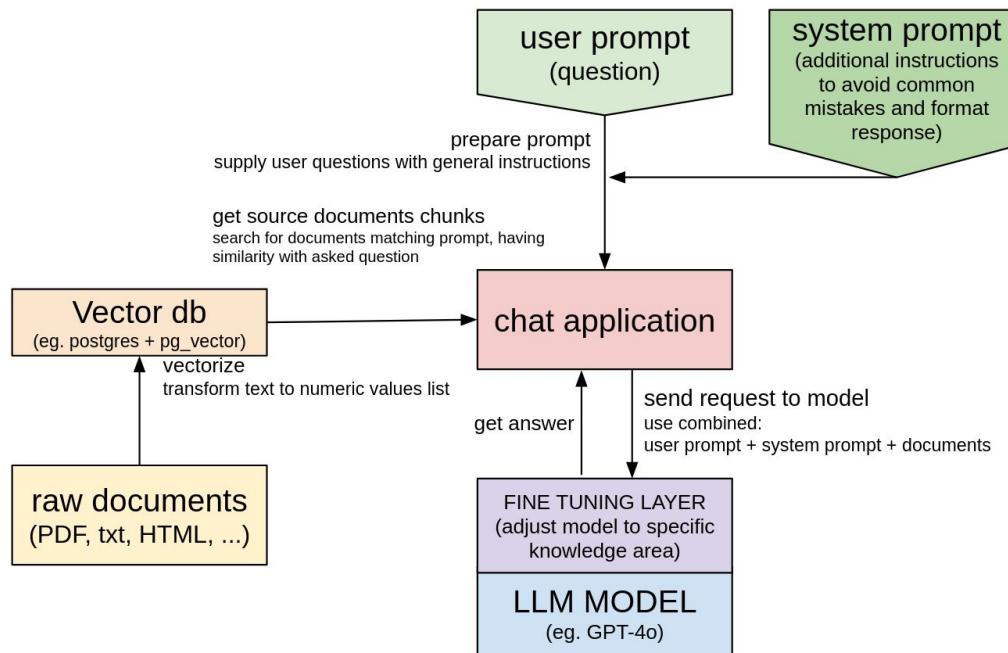
# RAG schema

What is Retrieval Augmented Generation (RAG) ?



# Application schema

## AI CHATBOT (LLM + RAG)



# Application

ConFoo.ca  
DEVELOPER CONFERENCE

Find answer in  
websites database

Is Michał Żarnecki programmer  
the same person as Michał  
Żarnecki audio engineer.

Generate answer

WIKIPEDIA  
Wolna encyklopedia

Search Wikipedia

Search

Michał Żarnecki [edit]

Article Discussion Read Edit Edit source code View History Tools

Beginning Curriculum vitae Filmography Awards and nominations Footnotes

Michał Żarnecki (born November 12, 1946 in Warsaw , died November 21, 2016 , there [1]) – Polish cinematographer and sound director.

Curriculum vitae [edit | edit code]

Winner of the Award for sound at the Polish Feature Film Festival in Gdynia and five times nominated for the Polish Film Award, the Eagle in the category of best sound. Professor at the Department of Sound Engineering at the Fryderyk Chopin University of Music in Warsaw .

Michał Żarnecki

Date and place of birth	November 12, 1946 Warsaw
Date and place of death	November 21, 2016 Warsaw
Occupation, occupation	sound operator



Michał Żarnecki Portfolio

I'm a programmer and lecturer. My work is related to programming in Python/PHP/JavaScript and designing systems and solutions related to AI/machine learning, data mining, big data and natural language processing.

LinkedIn Github

CompanyHouse Chat about company information

Our chatbot helps you answer questions about company history and reality. Ask a question and the chatbot will provide you with precise information based on official documents.

Find the company you were looking for:

Currently selected company: ACDI Abengesellschaft (registered)

Chatbot

Get instant, accurate answers Supported by official documents Relieve info from authorized sources

Start chat now

Our language models are powerful, but check important limitations.

Source document:

- Chronological commercial register extract

when last time company changed address

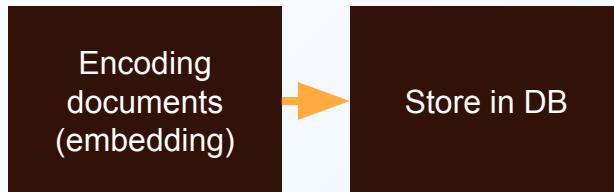
Address: According to the available documents, ACDI Abengesellschaft changed its business address on July 1, 2021. The new address is: Auf der Grün 10, 52070 Aachen.

events lectures projects

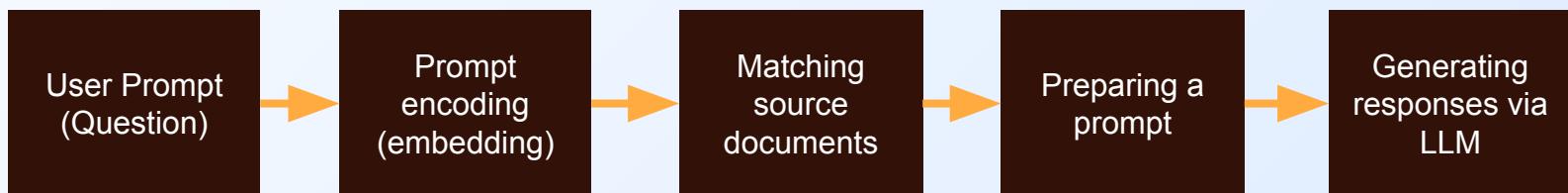
Cassandra Docker Big Data Tanglechain regular expression Elasticsearch API Server administration ScikitLearn LLM Pandas Vector DB Ubuntu Llama3 Android RAG Jenkins unsupervised ML JQuery Streamlit data science Kibana TensorFlow supervised ML NLP AWS npm PHP Spacy Neo4j reinforcement learning DBMS MongoDB PostgreSQL time series Machine Learning Data mining gulp NER Symfony Nginx Laravel TextBlob Pytorch JavaScript NumPy CentOS Node.js Neural networks Elasticsearch VUE nodejs

# Execution flow

## 1. Import documents



## 2. Generate answers



# Website dataset



Michał Żarnecki [edytuj]  
Artykuł Dyskusja Czytaj Edytuj Edytuj kod źródłowy Wyświetl Spis treści [ukryj]  
Początek Życiorys Filmografia Nagrody i nominacje Przypisy Michał Żarnecki (ur. 12 listopada 1946 w Warszawie, zm. 21 listopada 2016, tamże<sup>[1]</sup>) – polski operator i reżyser dźwięku.  
Życiorys Laureat Nagrody za dźwięk na Festiwalu Polskich Filmów Fabularnych w Gdyni oraz pięciokrotnie nominowany do Polskiej Nagrody Filmowej, Orzel w kategorii najlepszy dźwięk.

**Michał Żarnecki Portfolio**  
I'm a programmer and lecturer. My work is related to programming in Python/PHP/Javascript and designing systems and solutions related to AI/machine learning, data mining, big data and natural language processing.

LinkedIn Github

Chatbot™  
Give instant, accurate answers  
Supported by official documents  
Reliable site from authorized sources  
Get started now!

Our language models are powered by AutoGPT-3.0

Chat about company information  
Our dataset helps you understand about company details and history. Ask a question and the dataset will provide you with precise information about the company.

Find out more about our service  
Looking for companies...  
Currently selected company: Alibi Admangraphsoft (registered)

events lectures projects

Cassandra Docker MySQL regular expression Redis API Server administration SkicitLearn LLVM Pandas Vector DB Ubuntu Linux Android RAG Jenkins unsupervised ML Jquery Streamlit data science Kibana NLP Java supervised ML NLP AWS npm PHP Spacy Neo4j reinforcement learning DBMS MongoDB PostgreSQL time series Machine Learning Data mining NLP NER Symphony Nginx Laravel Textblob PyTorch JavaScript NumPy TensorFlow Neural networks Elasticsearch Node.js

HETUL MEHTA - UPDATED 3 YEARS AGO 84 New Notebook Download (2 MB) ::

## Website Classification

classify website URLs to different categories

Data Card Code (10) Discussion (2) Suggestions (0)

### About Dataset

#### Context

This dataset was created by scraping different websites and then classifying them into different categories based on the extracted text.

#### Content

Below are the values each column has. The column names are pretty self-explanatory.  
website\_url: URL link of the website.  
cleaned\_website\_text: the cleaned text content extracted from the

Usability 10.00  
License CC0: Public Domain  
Expected update frequency Annually  
Tags

<https://www.kaggle.com/datasets/hetulmehta/website-classification>

# Vector database

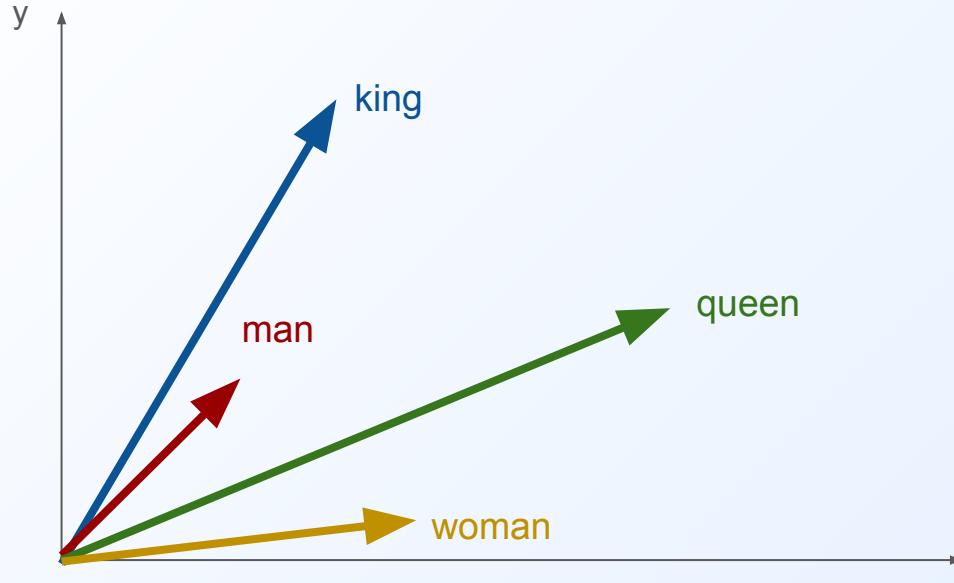
 PostgreSQL



pg\_vector



# Coding text to vector form



king [2, 4]

queen [5, 3]

...

king - man + woman  $\approx$  queen

I walked - I am walking  $\approx$  I flew - I am flying

we are - we  $\approx$  I am - I



# Coding text to vector form

```
model['beaver']
```

```
array([ 0.12233 , -0.40965 ,  0.26148 , -0.51868 , -0.60459 ,
       -0.84374 , -0.11416 ,  0.89577 , -0.23875 , -0.73708 ,
      -1.0004 , -0.098071 ,  0.18746 ,  0.14397 ,  0.47586 ,
       0.40886 ,  0.87898 ,  0.18226 , -0.32419 ,  0.54662 ,
      -0.46989 ,  0.90234 , -0.34768 ,  0.10074 ,  0.35826 ,
       0.31507 , -0.65955 , -0.042604 ,  0.010843 ,  0.050955 ,
       0.31938 , -0.10816 ,  0.22426 , -0.22073 ,  0.15903 ,
       0.16886 , -0.035559 ,  0.35242 ,  0.73245 ,  0.29862 ,
      -0.26791 , -0.09938 , -1.1552 ,  0.14245 ,  0.28243 ,
       0.50401 ,  0.27086 , -0.11168 , -0.2424 ,  0.29003 ,
      -1.3194 , -0.19863 ,  0.27979 ,  0.12743 , -0.5687 ,
      -1.1821 ,  0.15884 ,  0.0027925,  0.10209 , -0.10743 ,
       0.084864,  0.5932 ,  0.96694 ,  0.79069 ,  0.59454 ,
      -0.061516,  0.25304 , -0.0038544,  0.17405 ,  0.08916 ,
      -0.82945 , -0.56059 , -0.29723 ,  0.39116 ,  0.14941 ,
      -0.22372 , -0.3033 ,  0.12273 , -0.58946 ,  0.53444 ,
      -0.39859 ,  0.37937 , -0.2814 , -0.19535 ,  0.35982 ,
       0.24495 , -0.15736 , -0.45703 ,  0.71712 ,  0.75183 ,
      -0.43934 , -0.49642 ,  0.2353 , -0.039732, -0.47302 ,
      -0.15553 , -0.11614 ,  0.59744 ,  0.15876 ,  0.33116 ],
      dtype=float32)
```

# Coding text to vector form

```
model ['Canadian']
```

```
array([ 0.13072 , 0.0074814, 0.68605 , 0.22263 , 0.77077 ,  
       -0.63863 , -0.37411 , -0.74478 , -0.85476 , -0.12231 ,  
       -0.3389 , 0.073336 , 0.26157 , -0.02053 , -0.41198 ,  
       -0.035443 , 0.36547 , -0.61157 , -0.65216 , 0.27851 ,  
       0.31987 , 0.52021 , 0.66327 , -0.37198 , 0.15594 ,  
       -0.33592 , 0.66181 , -1.1405 , 0.7163 , 0.61639 ,  
       -0.42617 , 1.0608 , -0.54069 , -0.31184 , 0.2594 ,  
       -0.3142 , -0.50796 , 0.60521 , -0.023711 , -0.15069 ,  
       0.2187 , -0.1136 , -0.27597 , 0.37843 , 0.44209 ,  
       0.37464 , -0.46858 , -0.85882 , -0.15925 , -0.43794 ,  
       0.025459 , 0.18602 , 0.16313 , -0.011162 , -0.31522 ,  
       -1.496 , -0.40469 , -0.06208 , 1.9206 , 0.49531 ,  
       0.62159 , 0.25132 , 0.11749 , 0.49945 , 0.80494 ,  
       -0.18877 , -0.58738 , 0.41344 , 0.48951 , 0.28319 ,  
       -0.024222 , -0.090607 , -0.3953 , 0.6492 , -0.10056 ,  
       0.73393 , 0.76385 , 0.56781 , -1.0193 , -0.37452 ,  
       0.18781 , 0.2887 , 1.0299 , 0.43652 , -0.36852 ,  
       -0.75035 , 0.36968 , 0.50779 , 0.54883 , -0.27337 ,  
       -0.094371 , -0.75787 , -0.64647 , -0.002178 , -0.69057 ,  
       0.42273 , -0.45127 , 0.26008 , 0.20914 , -0.30791 ],  
      dtype=float32)
```

# Coding text to vector form

```
(model['Canadian'] + model['beaver']) / 2
```

```
array([ 0.126525 , -0.2010843 , 0.47376502, -0.14802499, 0.08309001,
       -0.74118495, -0.244135 , 0.075495 , -0.546755 , -0.42969498,
       -0.66964996, -0.0123675 , 0.224515 , 0.06172 , 0.03194 ,
       0.1867085 , 0.622225 , -0.214655 , -0.48817497, 0.412565 ,
       -0.07501 , 0.711275 , 0.157795 , -0.13562 , 0.2571 ,
       -0.010425 , 0.00112998, -0.59155196, 0.3635715 , 0.3336725 ,
       -0.053395 , 0.47631997, -0.158215 , -0.266285 , 0.20921502,
       -0.07267001, -0.2717595 , 0.47881502, 0.3543695 , 0.07396499,
       -0.024605 , -0.10649 , -0.715585 , 0.26044 , 0.36225998,
       0.439325 , -0.09886001, -0.48525 , -0.200825 , -0.073955 ,
       -0.64697045, -0.006305 , 0.22146001, 0.058134 , -0.44196 ,
       -1.33905 , -0.122925 , -0.02964375, 1.011345 , 0.19394 ,
       0.35322702, 0.42226002, 0.542215 , 0.64507 , 0.69974 ,
       -0.12514299, -0.16717 , 0.2047928 , 0.33178002, 0.186175 ,
       -0.426836 , -0.3255985 , -0.34626502, 0.52018 , 0.024425 ,
       0.255105 , 0.23027499, 0.34527 , -0.80438 , 0.07995999,
       -0.10539 , 0.334035 , 0.37425 , 0.120585 , -0.00434999,
       -0.2527 , 0.10615999, 0.02538002, 0.632975 , 0.23922999,
       -0.2668555 , -0.627145 , -0.205585 , -0.020955 , -0.581795 ,
       0.1336 , -0.283705 , 0.42876 , 0.18395 , 0.01162501],
      dtype=float32)
```

# Coding text to vector form

```
class TextEncoder extends AbstractGPTAPIClient implements StageInterface
{
    private string $embeddingModel = 'text-embedding-ada-002';

    public function getEmbeddings(string $document): string
    {
        $response = $this->client->embeddings()->create([
            'input' => $document,
            'model' => $this->embeddingModel
        ]);
        return json_encode($response->embeddings[0]->embedding);
    }
}
```

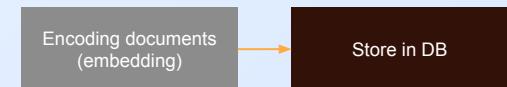


# Upload documents

```
class DocumentLoader extends AbstractDocumentRepository
{
    public function loadDocuments(): void
    {
        $path      = __DIR__ . '/../documents';
        $files = array_diff(scandir($path), array('.', '..'));
        foreach($files as $file) {
            $document = file_get_contents($path . '/' . $file);

            #load documents to postgresql database
            $responseDocument = $this->textEncoder->getEmbeddings($document);

            $this->insertDocument($document, $responseDocument);
        }
    }
}
```



# Upload documents

```
CREATE EXTENSION IF NOT EXISTS vector;  
  
CREATE TABLE IF NOT EXISTS document (  
    id serial PRIMARY KEY,  
    name text NOT NULL,  
    embedding vector,  
    text text,  
    chunk_id integer,  
    metadata JSONB,  
    created_at timestampz DEFAULT now()  
);
```



# Pipeline

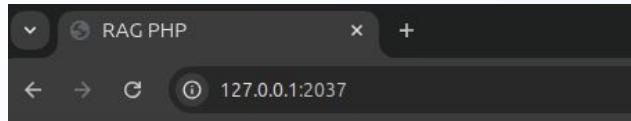
```
$promptResolver = new PromptResolver();
$textEncoder = new TextEncoder();
$documentProvider = new DocumentProvider();
$ragPromptProvider = new RAGPromptProvider();
$generatedTextProvider = new GeneratedTextProvider();
	payload = new Payload();

$pipeline = (new Pipeline(new FingersCrossedProcessor()))
    ->pipe($promptResolver) //get prompt from POST or CLI
    ->pipe($textEncoder) //get embeddings for prompt
    ->pipe($documentProvider) //find documents with similarity to prompt
    ->pipe($ragPromptProvider) //combine prompt with matched source documents
    ->pipe($generatedTextProvider); //get API response

$response = $pipeline->process($payload);
```



# User prompt



Find answer in  
websites database  
using LLM

Is Michał Żarnecki programmer  
the same person as Michał  
Żarnecki audio engineer.

Generate text



# Coding text to vector form

```
class TextEncoder extends AbstractGPTAPIClient implements StageInterface
{
    private string $embeddingModel = 'text-embedding-ada-002';

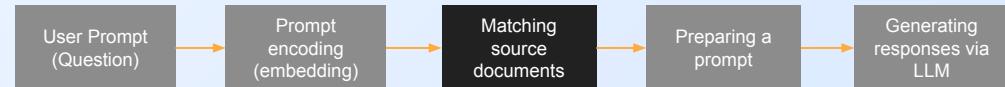
    public function getEmbeddings(string $document): string
    {
        $response = $this->client->embeddings()->create([
            'input' => $document,
            'model' => $this->embeddingModel
        ]);
        return json_encode($response->embeddings[0]->embedding);
    }

    /**
     * @param Payload $payload
     * @return Payload
     */
    public function __invoke($payload)
    {
        return $payload->setEmbeddingPrompt($this->getEmbeddings($payload->getPrompt()));
    }
}
```

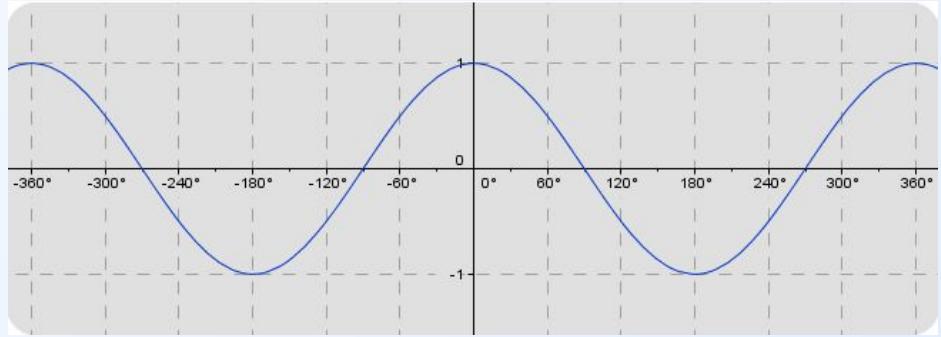
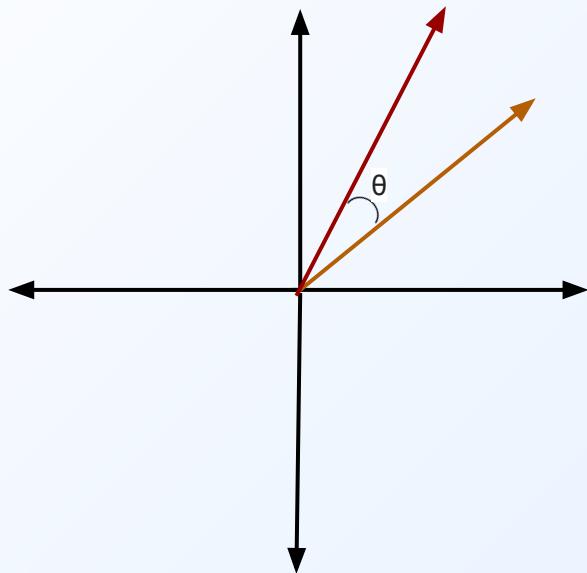


# Semantic search

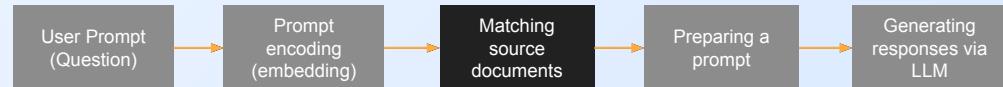
```
SELECT text FROM document order by embedding <=>  
'[-0.0014472235,-0.0001540061,0.0052023693,...]' DESC limit 10;
```



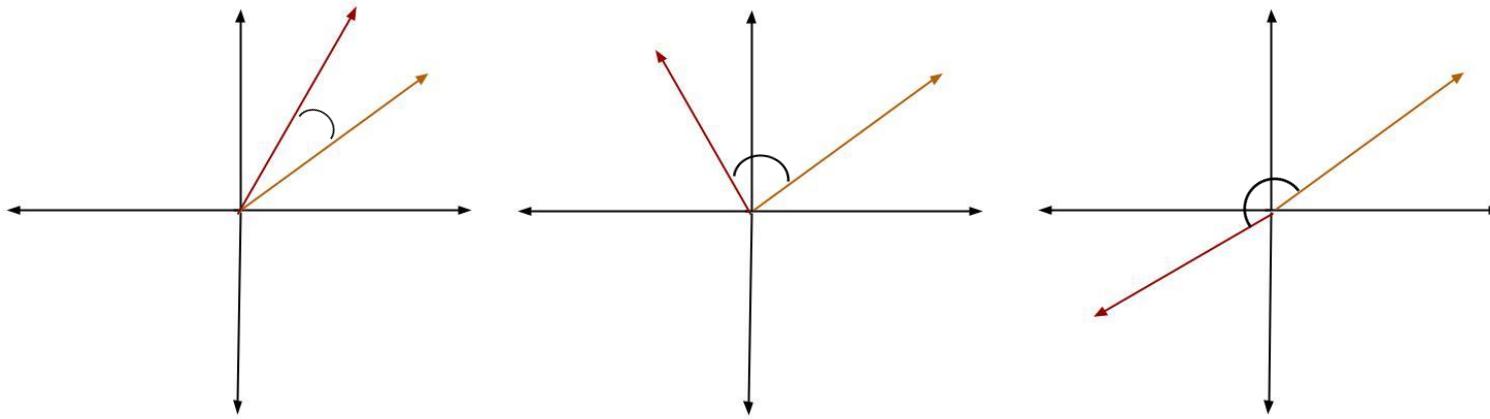
# Cosine similarity



$$\cos \theta = \frac{\vec{a} \cdot \vec{b}}{\|\vec{a}\| \cdot \|\vec{b}\|}$$



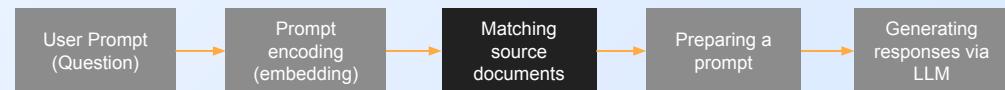
# Cosine similarity



angle close to  $0^\circ$   
cosine close to 1  
texts are similar

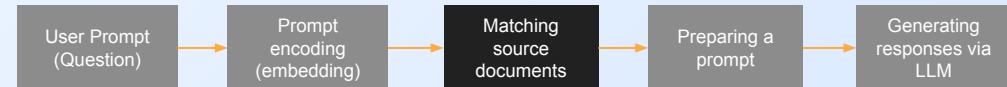
angle close to  $90^\circ$   
cosine close to 0  
texts are different

angle close to  $180^\circ$   
cosine close to -1  
texts are opposite



# Retrieve matching documents

```
class DocumentProvider extends AbstractDocumentRepository implements StageInterface
{
    public function getSimilarDocuments(string $embeddingPrompt): array
    {
        $stmt = $this->connection->prepare("SELECT text from document order by
embedding <=> :embeddingPrompt DESC limit 10;");
        $stmt->execute(['embeddingPrompt' => $embeddingPrompt]);
        return $stmt->fetchAll();
    }
}
```



# Matching algorithms

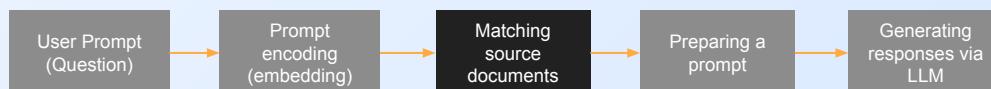
```
private function getQueryForL2Distance(int $limit = 10): string
{
    return "SELECT name, text from document order by
embedding <-> :embeddingPrompt limit {$limit};";
}

private function getQueryForCosineDistance(int $limit = 10): string
{
    return "SELECT name, text from document order by
1 - (embedding <=> :embeddingPrompt) limit {$limit};";
}

private function getQueryForInnerProduct(int $limit = 10): string
{
    return "SELECT name, text from document order by
(embedding <#> :embeddingPrompt) * -1 limit {$limit};";
}
```

character level and token  
level text similarity scores  
don't consider meaning

Metric	Considers Direction?	Considers Magnitude?	Captures Semantic Meaning?
Cosine Similarity	Yes	No	Yes (ideal for embeddings)
L2 Distance	Yes	Yes	Not directly
Dot Product	Yes	Yes	Yes (if embeddings are normalized)



# Model



by DALL-E

# Model



**GPT4o**

API OPENAI



**MIXTRAL**

API / STANDALONE  
45 B params

**ANTHROPIC**

**CLAUDE 3.5**  
**Sonnet**

API AWS BEDROCK 175 B params



**DeepSeek R1**

API /STANDALONE  
671 B params

**Meta**

**LLAMA3.3**

API / STANDALONE

70 B params  
context ~240 pages, 400 words each



**Gemini 2.0 Flash  
Experimental**

API GOOGLE

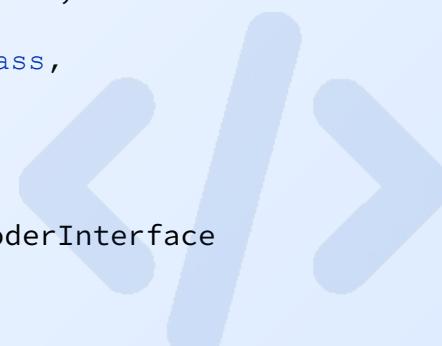
# Model

```
public function getGeneratedTextProvider(string $model): GeneratedTextProviderInterface
{
    $model = strtolower($model);
    $mapping = [
        'gpt-4o' => GeneratedTextFromGPTProvider::class,
        'claude-3.5' => GeneratedTextFromClaudeProvider::class,
        'deepseek' => GeneratedTextFromDeepSeekProvider ::class,
        'gemini2' => GeneratedTextFromGeminiProvider::class,
        'llama3.2' => GeneratedTextFromLocalLlama3Provider::class,
        'mixtral' => GeneratedTextFromMixtralProvider::class,
        'bielik' => GeneratedTextFromLocalBielikProvider ::class,
    ];
    ...
}

public function getEmbeddingsService(string $model): TextEncoderInterface
{
    $model = strtolower($model);
    $mapping = [
        'gpt-4o' => Ada002TextEncoder::class,
        ...
    ];
}
```

require API key from model provider

served with ollama locally



# Prompt engineering + one shot learning



by DALL-E



# Prompt engineering + one shot learning

## ##### INPUT:

You are a **helpful AI assistant** with access to a set of websites content. Your role is to provide information and answer questions **based solely on these documents**.

You should respond **directly and concisely**, using the information contained within the documents **without quoting or revealing actual document content**. **Do not infer or guess** information that is not explicitly stated in the documents.

If a question relates to **information not present in the documents, state that the information is not available**. Your goal is to be helpful by providing factual and document-backed answers.

Here is question:

<QUESTION>

Source documents:

<DOCUMENTS>

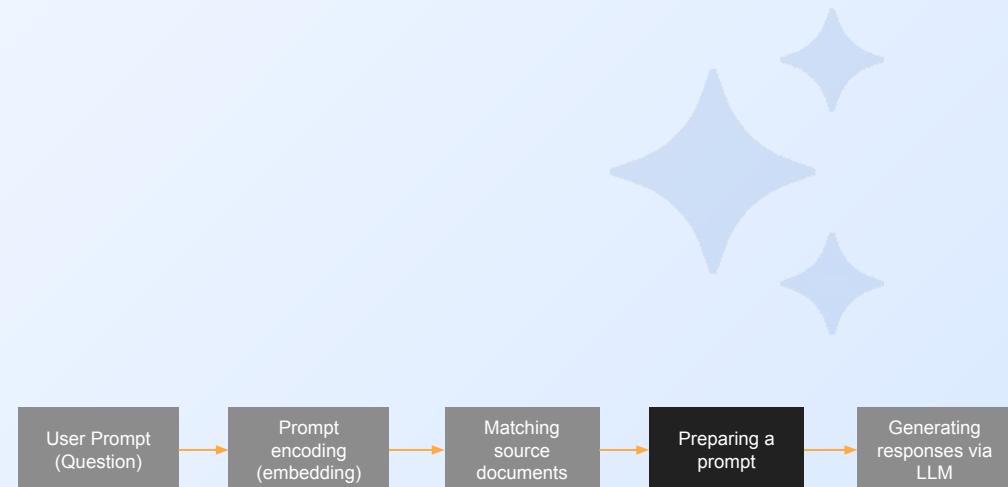
Example:

when receive question and content like below:

<WEBSITE CONTENT>

you should answer like below:

<EXPECTED ANSWER>

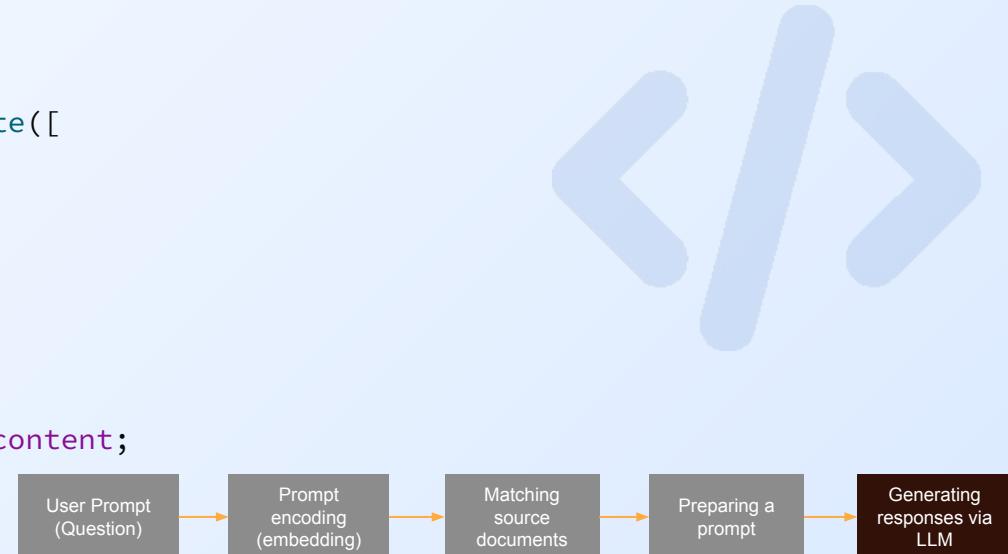


# API Communication

```
final class GeneratedTextFromGPTProvider extends AbstractGPTAPIClient
    implements StageInterface, GeneratedTextProviderInterface
{
    private string $model = 'gpt-4o';

    public function generateText(string $prompt, string $sourceDocuments): string
    {
        # prepare API input
        $input = $sourceDocuments . "\n\n##### INPUT: \n" . $prompt . "\n#####\n";
RESPONSE:\n";

        # get API response
        $response = $this->client->chat()->create([
            'model' => $this->model,
            'messages' => [
                [
                    'content' => $input,
                    'role' => 'user'
                ]
            ]
        ]);
        return $response->choices[0]->message->content;
    }
}
```



# Generate response

##### INPUT:

Is Michał Żarnecki programmer the same person as Michał Żarnecki audio engineer?



Michał Żarnecki  
Audio engineer

Michał Żarnecki (ur. 12 listopada 1946 w Warszawie, zm. 21 listopada 2016, tamże) – polski operator i reżyser dźwięku.

 Wikipedia  
[https://pl.wikipedia.org/w/index.php?title=Michał\\_Żarnecki&oldid=4500000](https://pl.wikipedia.org/w/index.php?title=Michał_Żarnecki&oldid=4500000)

**Michał Żarnecki – Wikipedia, wolna encyklopedia**

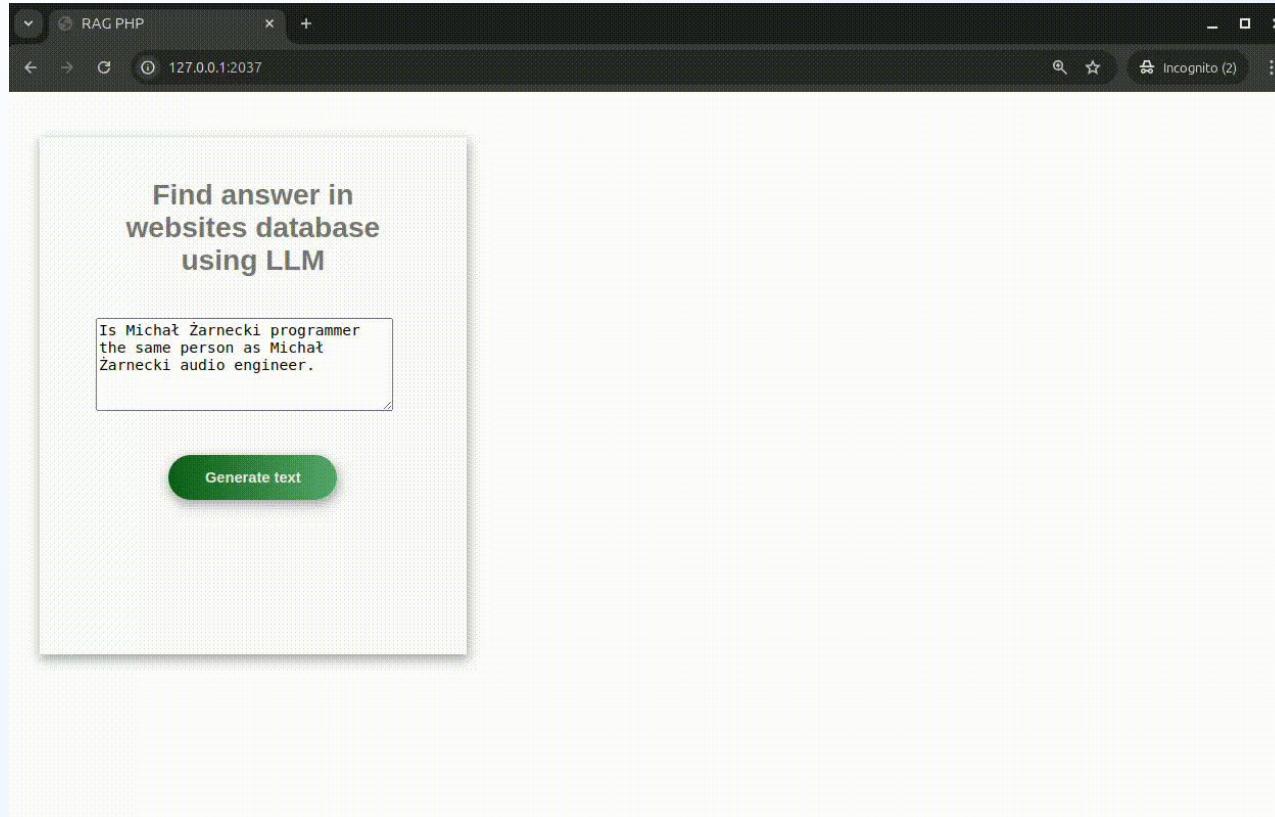
**Born:** November 12, 1946, Warsaw

**Died:** November 21, 2016, Warsaw

**Nominations:** Polish Academy Award for Best Sound

**Siblings:** Andrzej Żarnecki

# Demo



# Analyzing web content

## ##### INPUT:

Is Michał Żarnecki programmer the same person as Michał Żarnecki audio engineer?

## ##### RESPONSE:

**Based on the information provided, it appears that Michał Żarnecki the programmer and Michał Żarnecki the audio engineer are not the same person.**

Here's why:

### 1. \*\*Fields of Expertise\*\*:

- Michał Żarnecki, the audio engineer, was a well-known operator and reżyser dźwięku (sound director) in Poland, with notable contributions to the Polish film industry, as mentioned in the Wikipedia excerpt.
- Michał Żarnecki, the programmer, has a portfolio focused on programming in Python, PHP, and JavaScript, with projects related to AI, machine learning, data mining, and software development.

### 2. \*\*Lifespan\*\*:

- Michał Żarnecki the audio engineer was born on November 12, 1946, and passed away on November 21, 2016.
- The projects listed in Michał Żarnecki the programmer's portfolio date from 2014 to 2016, which would be conflicting if he had passed away in 2016 and was actively working in those years.

### 3. \*\*Occupational Focus\*\*:

- The audio engineer has a career documented in film sound engineering and education.
- The programmer's career is centered around software development, mobile applications, ERP systems, and consulting in technology.

Given the distinct differences in their professional domains, timelines, and expertise, it is highly unlikely that they are the same individual

# Which computer is best?



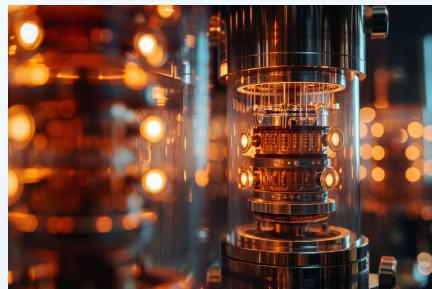
1. Raspberry Pi



2. Gaming PC with newest GPU



3. new MacBook



4. Quantum computer



5. Super computer in research center

# Evaluation criteria

-  **Correctness:** Is the answer accurate and free of mistakes?
-  **Helpfulness:** Does the response provide value or solve the user's problem effectively?
-  **Relevance:** Does the answer address the question accurately?
-  **Conciseness:** Is the answer free of unnecessary details?
-  **Clarity:** Is the language clear and understandable?
-  **Factual Accuracy:** Are the facts provided correct?

# Evaluation

	response	ROUGE recall	ROUGE precision	ROUGE F1	BLEU	GPT-4o criteria avg	correctness	helpfulness	relevance	conciseness	clarity	factual accuracy	insensitivity
Gemini 2 flash	No, the Michał Żarnecki...	0,45	0,7	0,55	0,4	99,9	5	5	5	5	5	5	5
Claude 3.5 Sonnet	No, these are two...	0,4	0,28	0,33	0,28	96,57	5	5	5	4	5	5	5
GPT-4o	...are two different individuals...	0,59	0,45	0,51	0,45	93,24	5	4	5	4	5	5	5
Bielik 2.3	Yes, Michał Żurawski...	0,23	0,3	0,26	0,22	33,3	2	1	1	3	2	1	5
Llama 3.2	I couldn't find...	0,22	0,3	0,25	0,21	79,92	4	3	4	5	4	4	5
Mistral 7B	no information to confirm...	0,18	0,57	0,28	0,07	79,92	3	3	5	5	5	3	5

# Analyzing web content

## ##### INPUT:

Is Michał Żarnecki programmer the same person as Michał Żarnecki audio engineer?

## ##### RESPONSE:

**Based on the information provided, it appears that Michał Żarnecki the programmer and Michał Żarnecki the audio engineer are not the same person.**

Here's why:

### 1. \*\*Fields of Expertise\*\*:

- Michał Żarnecki, the audio engineer, was a well-known operator and reżyser dźwięku (sound director) in Poland, with notable contributions to the Polish film industry, as mentioned in the Wikipedia excerpt.

- Michal Żarnecki, the programmer, has a portfolio focused on programming in Python, PHP, and JavaScript, with projects related to AI, machine learning, data mining, and software development.

### 2. \*\*Lifespan\*\*:

- Michał Żarnecki the audio engineer was born on November 12, 1946, and passed away on November 21, 2016.

- The projects listed in Michał Żarnecki the programmer's portfolio date from 2014 to 2016, which would be conflicting if he had passed away in 2016 and was actively working in those years.

### 3. \*\*Occupational Focus\*\*:

- The audio engineer has a career documented in film sound engineering and education.

- The programmer's career is centered around software development, mobile applications, ERP systems, and consulting in technology.

Given the distinct differences in their professional domains, timelines, and expertise, it is highly unlikely that they are the same individual

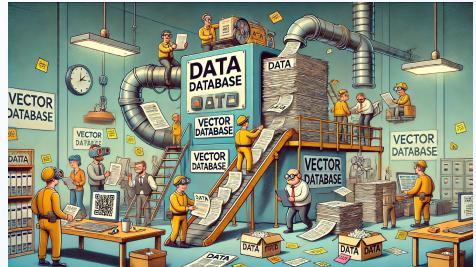


# How to implement RAG in a project?

# How to implement RAG in a project?

1

Export source data  
to vector db



2

Import and configure  
model or use  
API LLM



3

Prepare the prompt



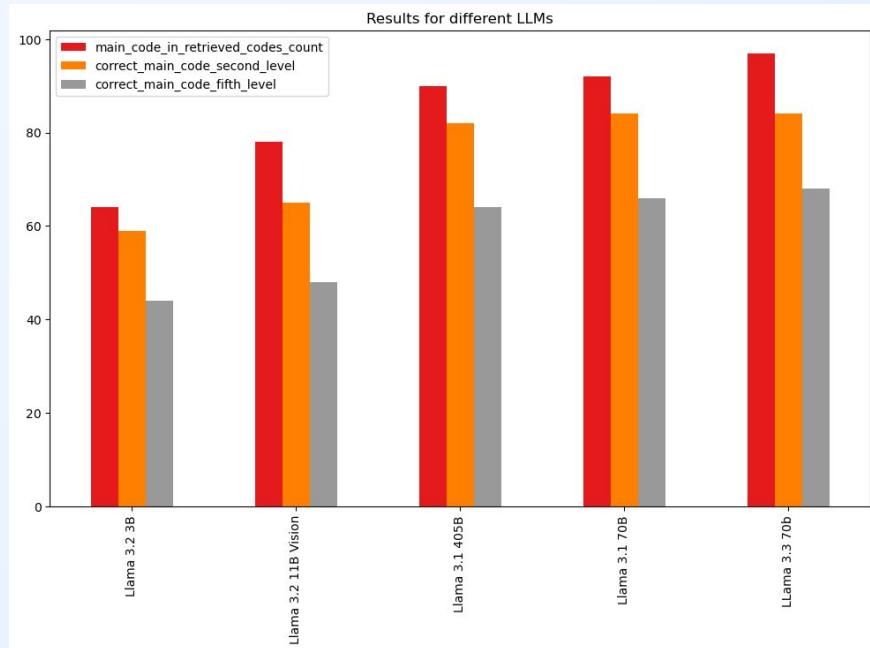
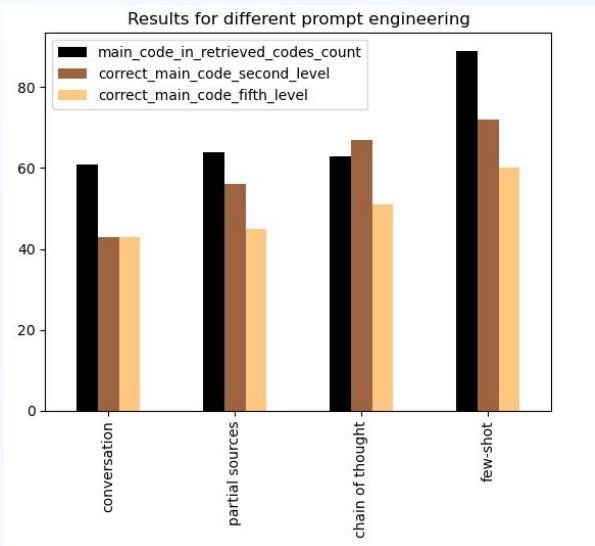
4

Optional:  
model fine-tuning



# Optimizations

- 🤖 Use different LLMs
- 🎯 Use different embeddings model
- ⚙️ Modify system prompt
- 📚 Use different number of retrieved documents
- ⚖️ Use reranking
- 🔍 Use different text matching algorithm



# Codebase expert & reviewer

ConFoo.ca  
DEVELOPER CONFERENCE



ConFoo.ca  
DEVELOPER CONFERENCE

## Replace complex business logic with Machine Learning models

Michał Żarnecki



<https://github.com/mzarnecki/ai-codebase-expert>

description: Person shown 2x on timeline Check why this person is shown twice and resolve the issue.

### Investigation Steps

#### 1. Check Data Retrieval Methods:

- `getCompanyManagers` and `getCompanySignatories`: These methods retrieve data about company managers and signatories. We need to ensure that these methods do not return duplicate entries. This can be done by checking the queries and filters applied in these methods.

#### 2. Check Data Addition Methods:

### Code Solution

Let's modify the `addCompanyManagersData` and `addCompanySignatoriesData` methods to ensure that duplicates are not added to the timeline.

```
private function addCompanyManagersData(  
    CompanyRecord $company,  
    CompanyTimelineModel $companyTimelineModel  
): void {  
    if ($persons = $this->getCompanyManagers($company)) {  
        $count = 0;  
        $addedPersons = []; // Track added persons to avoid duplicates  
        /** @var PersonFunctionCache $person */
```

# Thank you for watching

Michał Żarnecki

Contact: michal@zarnecki.pl

Github: <https://github.com/mzarnecki>

LinkedIn: <https://www.linkedin.com/in/michal-zarnecki-47219355/>

Add your data and start using RAG!

<https://github.com/mzarnecki/php-rag>



**ConFoo.CA**  
DEVELOPER CONFERENCE

Leave feedback and  
rate the presentation

