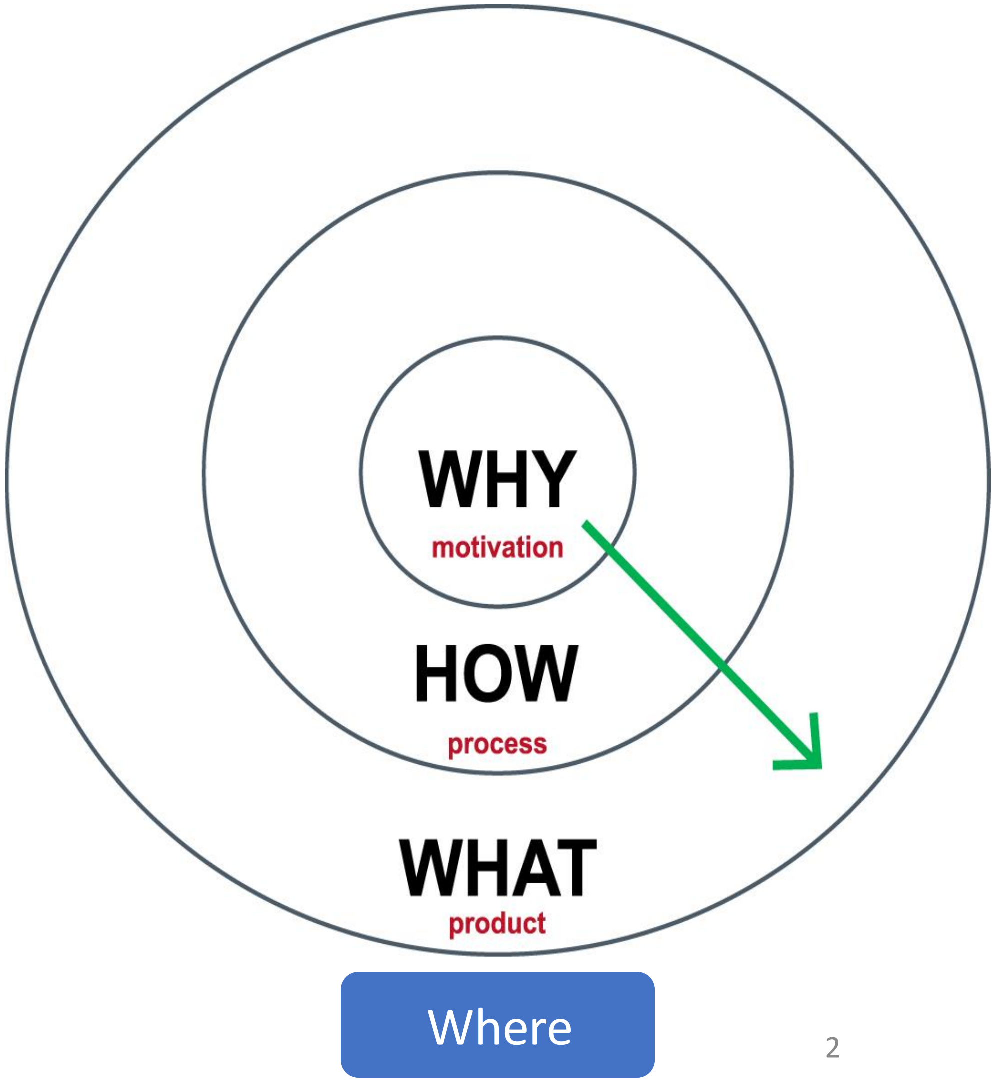
Promp compression:
Why, How, What and Where

Petr Lorenc

Agenda - Start with **WHY** …

- **WHY** we did it?
- **HOW** was it done?
- **WHAT** is there beneficial for you?
- **WHERE** can I try it?



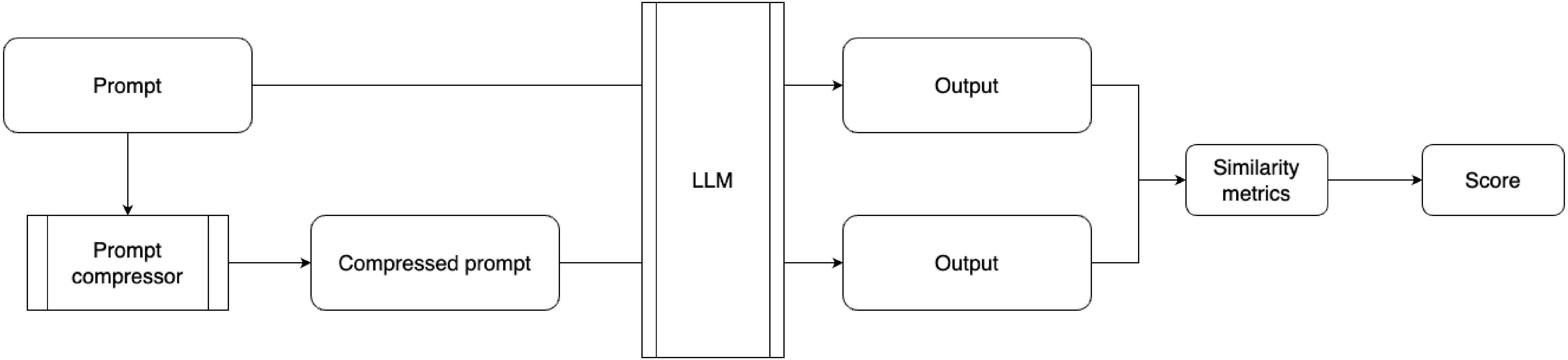
But **Why** is it important for **you**?

- Because we want to reduce a cost
  - It can reduce the token count = reduce the cost for the company

- Because we do not want to wait
  - It can speed up your prompt (less tokens for input)

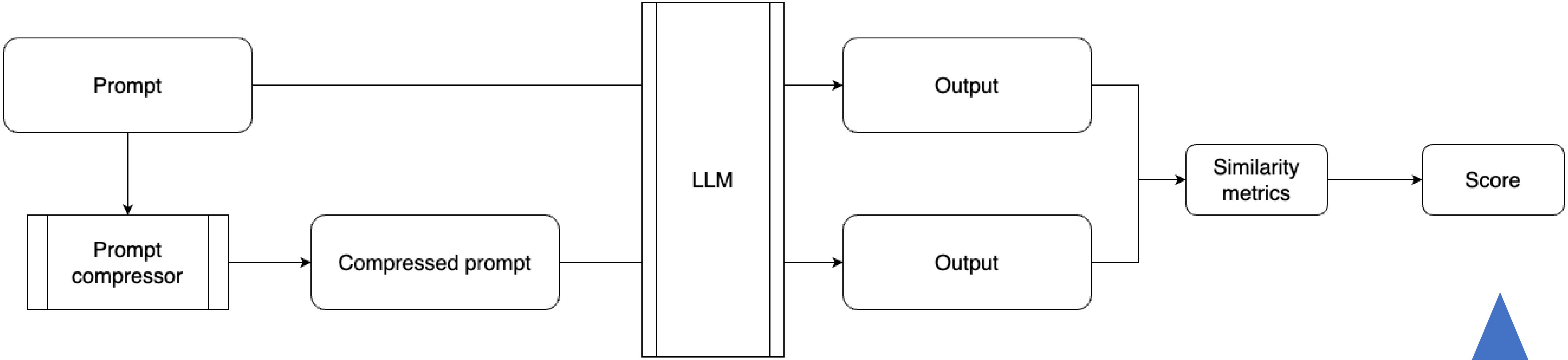
So **How** it was measured?

- We want the same output as without any prompt compression



Let’s start from the end …

- We want the same output as without any prompt compression



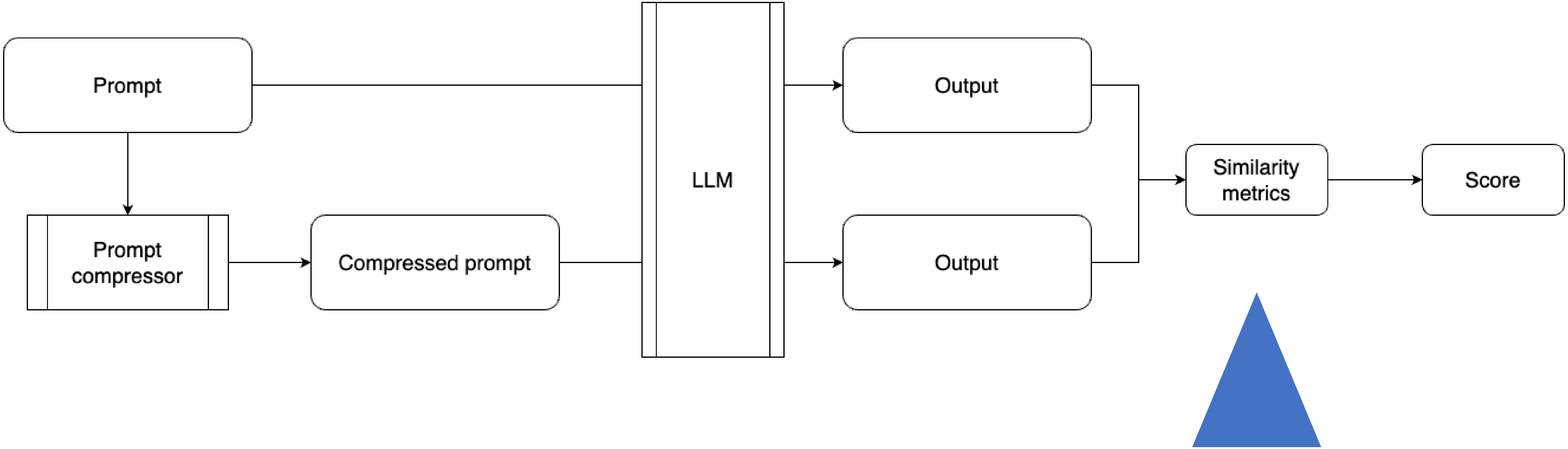
To get **Score** we need some test-data

- **Summarisation (similar to non-RAG applications)**:
  - Given a context, the task is to generate a summary that captures the main points of the document. This task aims to evaluate how compression affects the overall understanding of models on the input contexts.
  - https://huggingface.co/datasets/EdinburghNLP/xsum

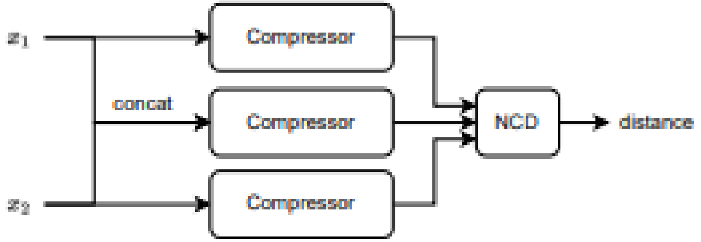
- **QA over Documents (similar to RAG applications):**
  - Given a document, the task is to answer given question.
  - We skip the Retrieval part (not a factor here)
  - https://huggingface.co/datasets/rag-datasets/rag-mini-bioasq

We have the dataset, now we need **metrics**

- We want the same output as without any prompt compression



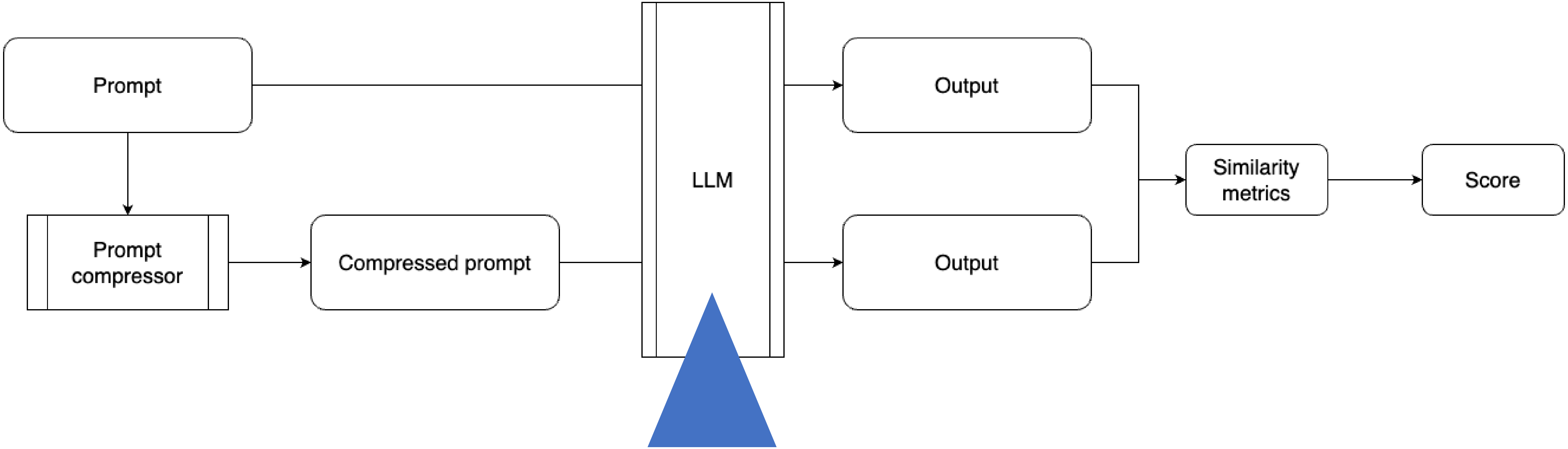
Selection of similarity metrics



- **BM25**
  - Based on tokens overlap
- **GZIP + NCD (Normalized Compression Distance)**
  - Based on difference between the compression rate of concatenated text and each part separately
- **ROUGE-L: Longest Common Subsequence (LCS)**
  - Based on tokens overlap
- **BERTscore F1**
  - Based on semantic similarity overlap of tokens (embeddings)
- **SentenceTransformers (SBERT) + cosine similarity**
  - Based on semantic overlap
- **OpenAI GPT4**
  - Based on “semantic” overlap + reasoning

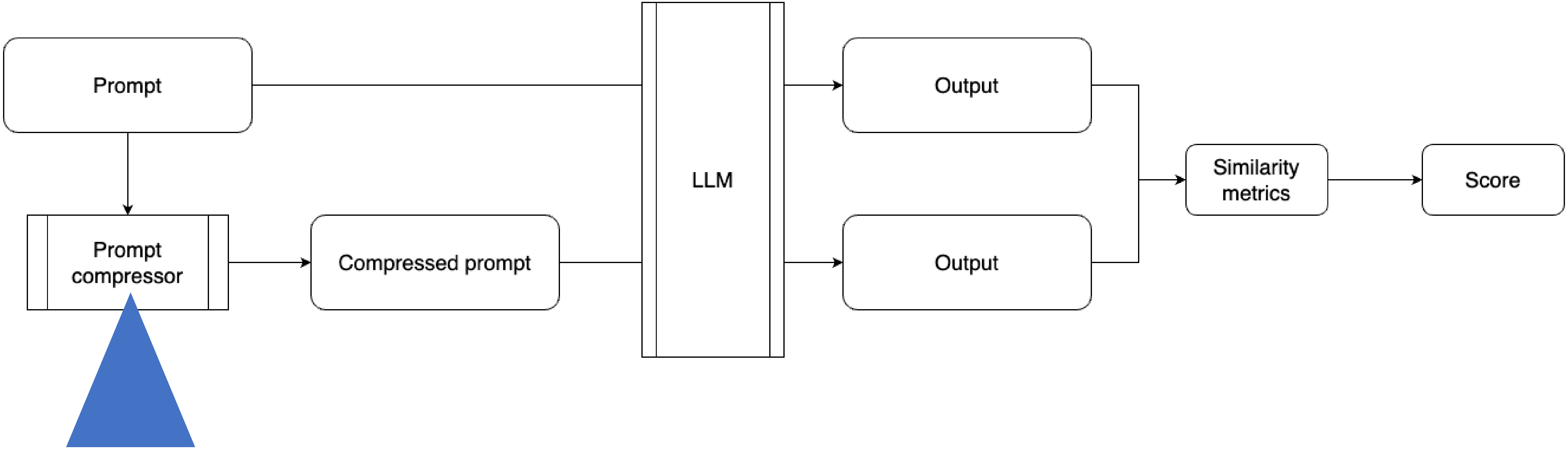
apiGPTeal family (Azure-based)

- We want the same output as without any prompt compression



And finally, we got to the compression part

- We want the same output as without any prompt compression



How it was measured?

- **Baselines**
  - Ask GPT4 to compress the prompt
  - Randomly remove 50% (chosen artificially because of other approaches)
- **Used Approaches**
  - LLMLingua
  - LongLLMLingua
  - LLMLingua 2

  - Selective Context
  - Prompt Optimizer

  - GPtrim

How it was measured?

- **Baselines**
  - Ask GPT4 to compress the prompt
  - Randomly remove 50% (chosen artificially because of other approaches)
- **Used Approaches**
  - LLMLingua
  - LongLLMLingua
  - LLMLingua 2
  - Selective Context
  - Prompt Optimizer
  - Gptrim

remove additional information

How it was measured?

- **Baselines**
  - Ask GPT4 to compress the prompt
  - Randomly remove 50% (chosen artificially because of other approaches)
- **Used Approaches**
  - LLMLingua
  - LongLLMLingua
  - LLMLingua 2

Assume well trained LLM model (with perplexity output)

Lower perplexity = lower contribution to the prompt

Lower perplexity = lower surprise of next token

I am hap.. -> …py VS …tic

  - Selective Context
  - Prompt Optimizer

  - Gptrim

Remove stop-words (the, a, …)

Stemming (running -> run)

What was measured?

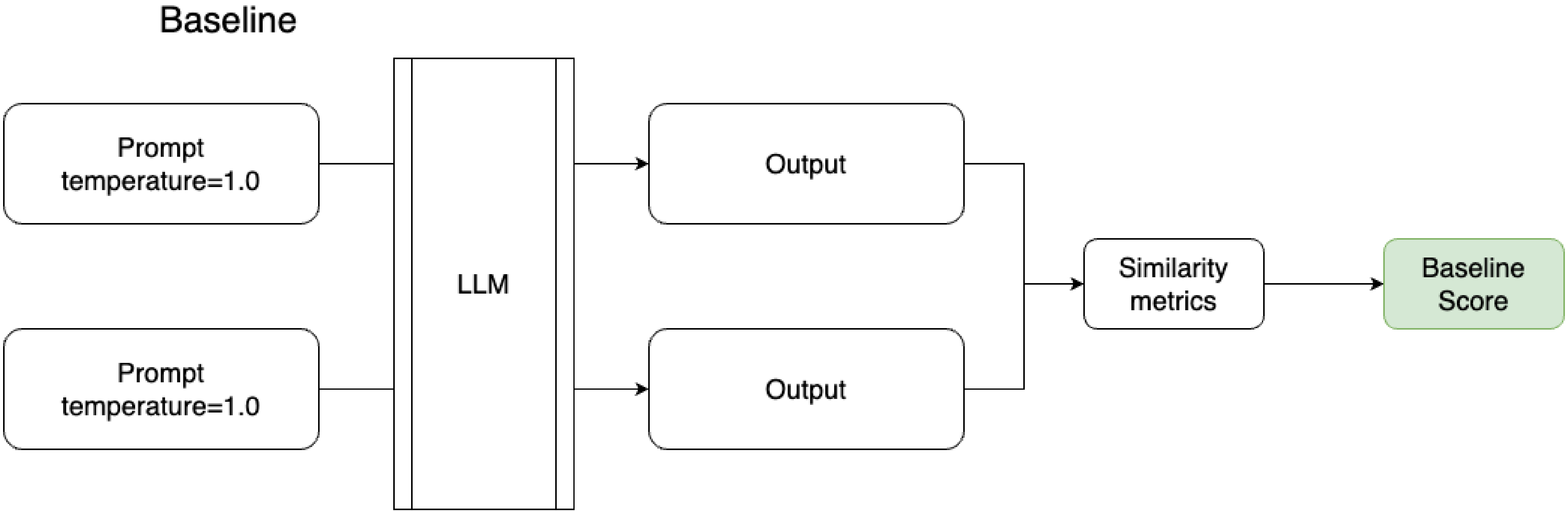
GPT 4 turbo	GPT 4	Random	LongLLMLingua	LLMLingua	LLMLingua 2	GPTrim
BM25	7.21	5.54	6.41	6.82	**11.86**	**10.9**
GZIP	0.53	0.58	0.55	0.54	**0.44**	**0.39**
ROUGE	0.53	0.59	0.56	0.559	**0.44**	**0.38**
BERTScore F1	0.112	0.127	0.11	0.117	**0.08**	**0.07**
SBert	0.100	0.117	0.10	0.116	**0.07**	**0.06**
GPT 4 as Judge	0.22	0.246	0.241	0.253	**0.14**	**0.117**
Latency	19.69 s	0.0025s	7.0732 s	5.3 s	**0.2333 s**	**0.0096 s**
Compression	2690 → 249 (10%)	2690 → 1322 (50%)	2690 → 1223 (45%)	2690 → 828 (30%)	**2690 → 1388 (51%)**	**2690 → 2323 (86%)**
HW requirements	None	None	GPU - 17-21G	GPU - 17-21G ml.g5.xlarge	**GPU - 1-2GB model**	**None**

What was measured?

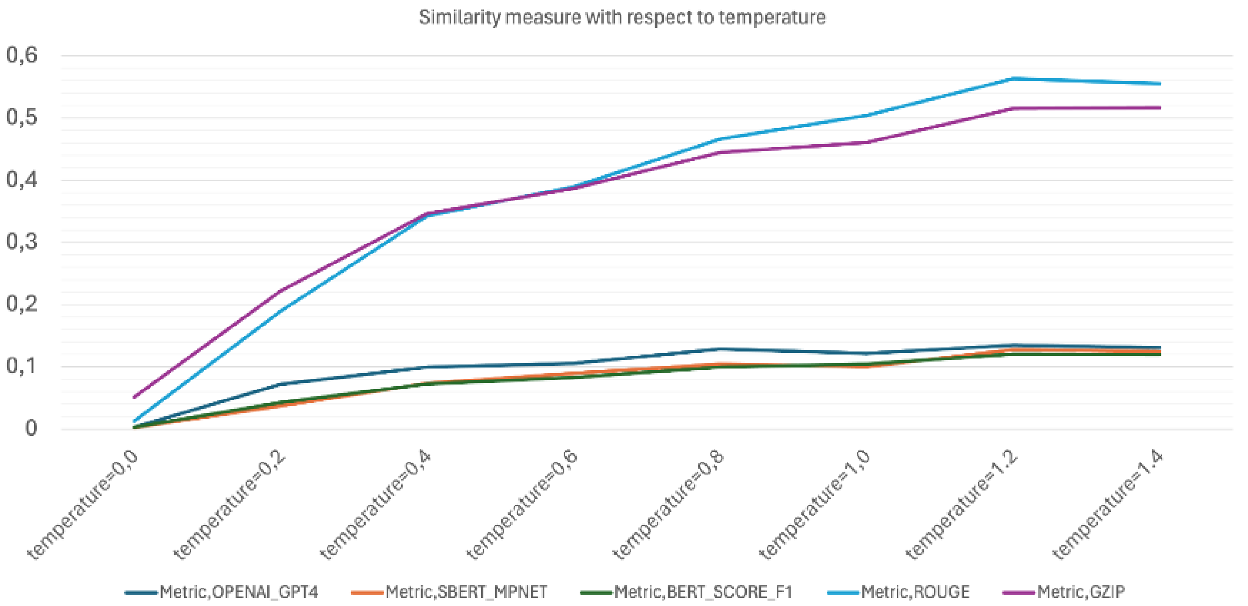
GPT 4 turbo	GPT 4	Random	LongLLMLingua	LLMLingua	LLMLingua 2	GPTrim
BM25	7.21	5.54	6.41	6.82	**11.86**	**10.9**
GZIP	0.53	0.58	0.55	0.54	**0.44**	**0.39**
ROUGE	[illegible]	[illegible]	[illegible]	[illegible]	**0.44**	**0.38**
BERTScore F1	[illegible]	[illegible]	[illegible]	[illegible]	**0.08**	**0.07**
SBert	[illegible]	[illegible]	[illegible]	[illegible]	**0.07**	**0.06**
GPT 4 as Judg	[illegible]	[illegible]	[illegible]	[illegible]	**0.14**	**0.117**
Latency	[illegible]	0.0025s	7.0732 s	3.3 s	**0.2333 s**	**0.0096 s**
Compression	2690 → 249 (10%)	2690 → 1322 (50%)	2690 → 1223 (45%)	2690 → 828 (30%)	**2690 → 1388 (51%)**	**2690 → 2323 (86%)**
HW requirements	None	None	GPU - 17-21G	GPU - 17-21G ml.g5.xlarge	**GPU - 1-2GB model**	**None**

Is it good result or not?

Is it good result or not?



RAG	T=0,0	T=0,2	T=0,4	T=0,6	T=0,8	T=1,0	T=1.2	T=1.4
BM25	16.98	15.69	14.71	14.51	13.90	**13.18**	12.32	11.69
GZIP	0.23	0.26	0.33	0.42	0.44	**0.47**	0.53	0.55
ROUGE	0.20	0.23	0.32	0.41	0.45	**0.46**	0.56	0.58
BERTScore	0.037	0.042	0.05	0.077	0.080	**0.089**	0.10	0.11
SBert	0.019	0.027	0.045	0.069	0.066	**0.062**	0.09	0.08
GPT 4	0.068	0.083	0.095	0.14	0.13	**0.14**	0.17	0.18



RAG	T=0,0	T=0,2	T=0,4	T=0,6	T=0,8	T=1,0	T=1.2	T=1.4
BM25 ↑	16.98	15.69	14.71	14.51	13.90	**13.18**	12.32	11.69
GZIP ↓	0.23	0.26	0.33	0.42	0.44	**0.47**	0.53	0.55
ROUGE ↓	0.20	0.23	0.32	0.41	0.45	**0.46**	0.56	0.58
BERTScore ↓	0.037	0.042	0.05	0.077	0.080	**0.089**	0.10	0.11
SBert ↓	0.019	0.027	0.045	0.069	0.066	**0.062**	0.09	0.08
GPT 4 ↓	0.068	0.083	0.095	0.14	0.13	**0.14**	0.17	0.18

GPT 4 turbo	LLMLingua 2	GPTrim	T=1,0
BM25	11.86	10.9	13.18 ↑
GZIP	0.44	0.39	0.47 ↓
ROUGE	0.44	0.38	0.46 ↓
BERTScore F1	0.08	0.07	0.089 ↓
SBert	0.07	0.06	0.062 ↓
GPT 4 as Judge	0.14	0.117	0.14 ↓

GPT 4o	LLMLingua 2	GPTrim	T=1,0
BM25	14.27	14.79	13.18 ↑
GZIP	0.466	0.425	0.47 ↓
ROUGE	0.453	0.404	0.46 ↓
BERTScore F1	0.095	0.0869	0.089 ↓
SBert	0.063	0.0563	0.062 ↓
GPT 4 as Judge	0.168	0.145	0.14 ↓

How it looks like in practise?

Many in **the list have been** found down old mine tunnels **or on** slag heaps where water and **even fire have had the** opportunity to work **up** novel com**pound**s.
It is another example, the researchers argue, of our **per**vasive influence on the planet.\n

=

the list have been or on even fire have had the upound**sIt** another per planet.

So **WHAT** is there beneficial for you?

- You can save the money spend ✓
  - Company pays for the tokens, so less tokens = less money

- You can reduce the latency of your prompt, but ⚖
  - it depends on the method
    - LLMLingua 2 is getting fast but still it is additional model
    - Gptrim has lower performance but it is super fast

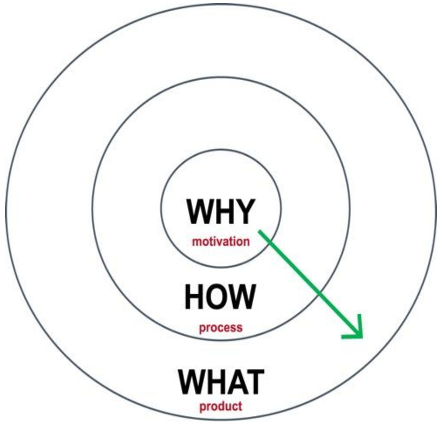
- You will get the similar output as having reasonble low temperature ⚖

Conclision - Start with **WHY …** End with answers

- **WHY** we did it?
  - Because we want work with longer context.

- **HOW** was it done?
  - By comparing output of uncompressed and compressed prompt

- **WHAT** is there beneficial for you?
  - It can save cost and potentionally speed up your repeating prompt


WHY
motivation
HOW
process
WHAT
product