

Neural Networks and Deep Learning

Lecture 12: Large language models in finance and prompt engineering





Prof. dr. Daniel Traian PELE

Bucharest University of Economic Studies (ASE), Department of Statistics and Econometrics
Romanian Academy, Institute for Economic Forecasting (IPE)
IDA Institute Digital Assets ▪ AI4EFin ▪ MSCA Digital Finance

Learning objectives

- Write the language-model objective, perplexity and the temperature of sampling, and measure them on central-bank speeches.
- Explain pre-training, instruction tuning and alignment from human preferences, with their loss functions.
- Turn a prompt into a classifier and measure zero-shot, few-shot and chain-of-thought prompts on labelled financial text, with confidence intervals.
- Build and evaluate retrieval-augmented generation (RAG) over real speeches, and measure hallucination.
- Backtest a language model used for the Value-at-Risk (VaR), and know the risks: leakage, cost, calibration.

Abbreviations

AI	artificial intelligence
API	application programming interface
ASE	Bucharest University of Economic Studies
BERT	bidirectional encoder representations from transformers
BM25	Best Matching 25 ranking function
BPE	byte-pair encoding
DPO	direct preference optimisation
ECB	European Central Bank
ECE	expected calibration error
ES	expected shortfall
EU	European Union
FinBERT	BERT model adapted to financial text
GARCH	generalised autoregressive conditional heteroskedasticity
GPT	generative pre-trained transformer
HS	historical simulation
IPE	Institute for Economic Forecasting, Romanian Academy
KL	Kullback–Leibler divergence
LLM	large language model
LoRA	low-rank adaptation
LR	likelihood ratio
MRR	mean reciprocal rank
RAG	retrieval-augmented generation
RLHF	reinforcement learning from human feedback
SFT	supervised fine-tuning
TF-IDF	term frequency–inverse document frequency
VaR	Value-at-Risk

Outline

- Next-token prediction and large language models
- Pre-training, instruction tuning and alignment
- Evaluation of prompt engineering on labelled data
- Retrieval-augmented generation on ECB speeches
- Language models as risk models: LLM-VaR
- Risks and governance

Text data in financial institutions

- **Markets**
  - News and social media move prices within minutes; sentiment is a signal
- **Central banks**
  - Speeches and minutes guide rate expectations; the European Central Bank (ECB) published 400 speeches since 2022 in our data
- **Credit and risk**
  - Loan files, annual reports, risk reports: reading them costs analyst hours
- **This lecture**
  - Open large language models (LLM) run on a laptop, measured on labelled data, never trusted on faith

A language model: next-token probabilities, trained by cross-entropy

- **In one sentence:** everything an LLM does, from answering questions to writing code, is repeated prediction of the next token.

$$p_\theta(x_1, \ldots, x_T) = \prod_{t=1}^{T} p_\theta(x_t \mid x_{<t})$$

$$\mathcal{L}(\theta) = -\frac{1}{T} \sum_{t=1}^{T} \log p_\theta(x_t \mid x_{<t}), \quad \mathrm{PPL} = e^{\mathcal{L}}$$

- **On ECB speeches** after July 2025, first 1,024 tokens: perplexity 14.71 (0.5B), 10.63 (1.5B), 9.43 (3B).

- **where**
  - x_t = token t
  - p_θ = softmax over the vocabulary at the last layer of a causal transformer (Lecture 7)
  - Perplexity = the effective number of equally likely next tokens.

- **In practice:** a lower perplexity on your own documents means the model 'speaks' your domain better.

(Bishop & Bishop, 2024, ch. 12)

Experiment yourself: notebook §1

Subword tokens and byte-pair encoding

- **In one sentence:** frequent words are one token, rare words and numbers are split; the model never sees letters, only token indices.

$$(a, b)^* = \arg\max_{(a,b)} \ \mathrm{count}(a\,b) \ \Rightarrow \ \text{merge } a\,b \to ab$$

- **where**
  - Repeated until the vocabulary is full: byte-pair encoding (BPE), 151 665 tokens in Qwen2.5
  - WordPiece, 30 522 tokens in a model of bidirectional encoder representations from transformers (BERT) adapted to finance (FinBERT).

- **Example** (20 tokens): - | Cash | flow | from | operating | activities | before | investments | was | EUR | | 0 | . | 8 | - | 1 | . | 2 | million | ..
- **Financial PhraseBank**: 1.35 Qwen tokens and 1.24 WordPiece tokens per word.

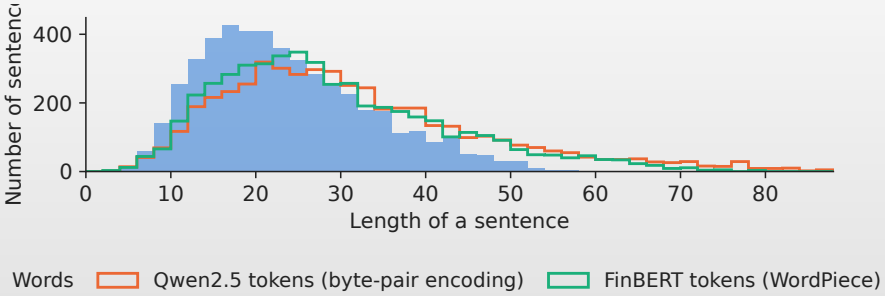
- **In practice:** cost and context limits are counted in tokens, not words: budget about 1.3 tokens per English word.

(Sennrich, Haddow & Birch, 2016)

Experiment yourself: notebook §1

Sentence length in words and in tokens: two tokenisers

- **In one sentence:** the same 4 838 sentences are a little longer in tokens than in words.



Source: Financial PhraseBank (Malo et al., 2014). Data: 4,838 sentences.

- Mean: 23.1 words, 31.1 Qwen tokens, 28.6 WordPiece tokens
  - An ECB speech has on average 4 143 tokens (longest 22 897).

NN_L12_tokens_per_sentence

Temperature reshapes the next-token distribution

- **In one sentence:** the model gives scores (logits); temperature decides how boldly we sample from them.

$$p_T(v \mid x_{<t}) = \frac{\exp(z_v/T)}{\sum_u \exp(z_u/T)}$$

$$H_T = -\sum_v p_T(v) \log p_T(v)$$

- **where**
  - z_v = logit of token v
  - $T \to 0$: always the most likely token (greedy)
  - $T > 1$: flatter, more diverse, more errors
  - H_T = entropy in nats.

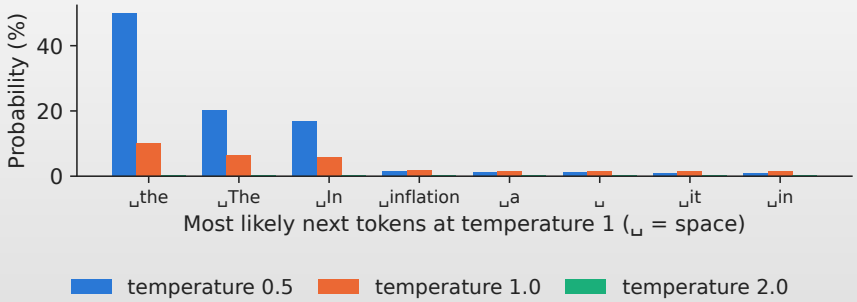
- **Prompt** from the ECB speech of 30 Oct 2026: "The inflation impact is limited: . . . "
- **Top token** "the": 49.9 % at $T = 0.5$, 10.1 % at 1, 0.4 % at 2
  - Entropy 1.74, 6.20, 10.29.

- **In practice:** for classification and extraction use temperature 0
  - Reserve high temperatures for brainstorming.

Experiment yourself: notebook §2

Next-token probabilities of a real model on a real sentence

- **In one sentence:** at low temperature almost all mass sits on one continuation; at high temperature unlikely tokens become plausible.



Source: European Central Bank, speeches dataset. Data: prompt from the ECB speech of 30 Oct 2026; model Qwen2.5-1.5B-Instruct.

NN_L12_next_token

Scaling laws: prediction loss as a power law of model size

- **In one sentence:** the loss falls as a power of the number of parameters and of training tokens.

$$\mathcal{L}(P, D) = E + \frac{A}{P^{\alpha}} + \frac{B}{D^{\beta}}$$

own estimate: $\log \mathcal{L} = a + b \log P, \quad \hat{b} = -0.100$

- **where**
  - P = parameters
  - D = training tokens
  - E = irreducible loss
  - $\alpha, \beta \approx 0.3$ (Hoffmann et al., 2022)
  - $\hat{b}$ from the three Qwen2.5 sizes on ECB speeches after July 2025.

- **Three points are not a law**: the slope is a description of our sample, not a forecast.
- **Compute-optimal**: about 20 training tokens per parameter.

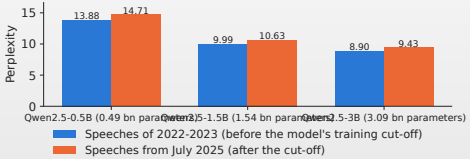
- **In practice:** a 3B model on a laptop costs nothing per call
  - A frontier model costs per token: measure the gain first.

(Kaplan et al., 2020; Hoffmann et al., 2022)

Experiment yourself: notebook §1

Training-data cut-off and the apparent ease of older speeches

- **In one sentence:** text the model may have seen in pre-training scores a lower perplexity than text written after it.



Source: European Central Bank, speeches dataset. Data: 40 + 40 speeches, first 1,024 tokens of each.

- 3B: 8.90 on 2022–2023 speeches vs 9.43 after July 2025
  - Lower for the older speeches at every size.
- Topics also changed
  - The gap is a symptom, not a proof, of memorisation.

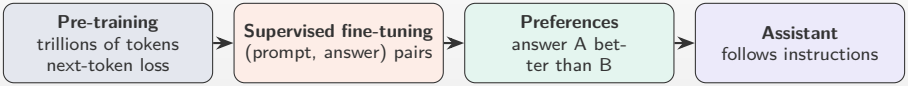
- **In practice:** evaluate an LLM on documents written after its training cut-off, or the score may be inflated by leakage.

NN_L12_perplexity_size

Outline

- Next-token prediction and large language models
- Pre-training, instruction tuning and alignment
- Evaluation of prompt engineering on labelled data
- Retrieval-augmented generation on ECB speeches
- Language models as risk models: LLM-VaR
- Risks and governance

Three training stages of a language-model assistant



- **Generative pre-trained transformer (GPT)**: stage 1 alone gives a model that continues text, not one that answers.
- **Supervised fine-tuning (SFT)** teaches the format
  - **reinforcement learning from human feedback (RLHF)** or **direct preference optimisation (DPO)** teaches which answers people prefer.
- **Qwen2.5-Instruct** (Qwen Team, 2024): 18 trillion pre-training tokens, then SFT and preference optimisation.

(Ouyang et al., 2022; Qwen Team, 2024)

Instruction tuning is next-token loss on the answer only

- **In one sentence:** the model sees prompt and answer, but is graded only on the answer tokens.

$$\mathcal{L}_{\text{SFT}}(\theta) = -\sum_{(\mathbf{x},\mathbf{y})\in\mathcal{D}} \sum_{t=1}^{|\mathbf{y}|} \log p_\theta\left(y_t \mid \mathbf{x}, \mathbf{y}_{<t}\right)$$

- **where**
  - $\mathbf{x}$ = instruction (with a chat template: system, user, assistant roles)
  - $\mathbf{y}$ = reference answer
  - $\mathcal{D}$ = tens of thousands to millions of pairs.

- **Low-rank adaptation (LoRA)**: train $\mathbf{W} = \mathbf{W}_0 + \mathbf{BA}$ with $\mathbf{B} \in \mathbb{R}^{d\times r}$, $\mathbf{A} \in \mathbb{R}^{r\times k}$, $r \ll d$
  - $\mathbf{W}_0$ frozen.
- **A** 4096×4096 **layer,** $r = 8$: 65,536 trainable weights instead of 16.8 million.

- **In practice:** a bank adapts an open model to its own documents with LoRA on one graphics card, keeping data in-house.

(Hu et al., 2022)

Alignment: reward models and direct preference optimisation

- **In one sentence:** people say which of two answers is better; the model is pushed towards preferred answers without drifting too far.

$$\Pr(y_w \succ y_l \mid x) = \sigma\big(r_\phi(x, y_w) - r_\phi(x, y_l)\big)$$

$$\max_{\theta} \ \mathbb{E}\big[r_\phi(x, y)\big] - \beta\, \mathrm{KL}\big(p_\theta \,\|\, p_{\mathrm{SFT}}\big)$$

$$\mathcal{L}_{\mathrm{DPO}} = -\log \sigma\left(\beta \log \frac{p_\theta(y_w|x)}{p_{\mathrm{SFT}}(y_w|x)} - \beta \log \frac{p_\theta(y_l|x)}{p_{\mathrm{SFT}}(y_l|x)}\right)$$

- **where**
  - y_w, y_l = preferred and rejected answers
  - r_ϕ = reward model (Bradley–Terry)
  - σ = logistic function
  - Kullback–Leibler (KL) penalty with weight β keeps the model close to the SFT model.

- **RLHF** fits r_ϕ, then optimises the policy
  - **DPO** skips the reward model.
- **Side effect**: aligned models sound confident
  - Confidence is not accuracy (calibration, below).

- **In practice:** alignment sets tone and safety, not financial knowledge: evaluate on your task.

(Ouyang et al., 2022; Rafailov et al., 2023)

The open models used in this lecture

Model	Parameters	Role here
Qwen2.5-0.5B-Instruct	0.49 bn	perplexity, zero-shot classification
Qwen2.5-1.5B-Instruct	1.54 bn	next-token probabilities, few-shot, embeddings
Qwen2.5-3B-Instruct	3.09 bn	best prompts, chain-of-thought, question answering, LLM-VaR
BAAI bge-small-en-v1.5	33 m	dense retrieval embeddings (384 dimensions)
Term frequency–inverse document frequency (TF-IDF) + logistic regression	–	classical benchmark trained on the labelled data

- All models run locally on a laptop; no text leaves the machine and no application programming interface (API) is paid.
- TF-IDF: word and word-pair counts weighted by their rarity across documents.

(Qwen Team, 2024; Xiao et al., 2023)

Outline

- Next-token prediction and large language models
- Pre-training, instruction tuning and alignment
- Evaluation of prompt engineering on labelled data
- Retrieval-augmented generation on ECB speeches
- Language models as risk models: LLM-VaR
- Risks and governance

Structure of a prompt

- **System**
  - Role and rules: "You are a careful financial analyst."
- **Instruction**
  - Task, label set, output format: "Answer with one word: positive, negative or neutral."
- **Examples** (optional)
  - k labelled input–answer pairs: few-shot
- **Input**
  - The sentence, tweet or document to process

(Brown et al., 2020; Wei et al., 2022)

- **Variants we test**
  - Zero-shot: instruction + input
  - Few-shot: 3 to 24 examples
  - Chain-of-thought: "explain, then answer"
  - Five paraphrases of the instruction
- **Rule of this lecture**
  - Every prompt is a model: it is scored on a held-out test set

A prompt as a classifier: probabilities of the label words

- **In one sentence:** read the probability of each label at the answer position: no text generation, no parsing, and a calibrated-looking score.

$$\hat{p}(c \mid x) = \frac{\sum_{v \in \mathcal{V}_c} p_\theta(v \mid \text{prompt}(x))}{\sum_{c'} \sum_{v \in \mathcal{V}_{c'}} p_\theta(v \mid \text{prompt}(x))}$$

$$\hat{c} = \arg\max_c \hat{p}(c \mid x)$$

- **where**
  - $\mathcal{V}_c$ = first tokens of the label word c in its spellings ("positive", " Positive", ...)
  - The prompt ends where the assistant's answer starts.

- **Fast**: one forward pass per item
  - The shared instruction and examples are encoded once (key–value cache).
- **Comparable**: the same macro-F1 and confidence intervals as any classifier.

- **In practice:** constrain the output space
  - Free text answers need parsing and fail silently.

(Zhao et al., 2021)

Experiment yourself: notebook §3

Two labelled data sets: financial news and finance tweets

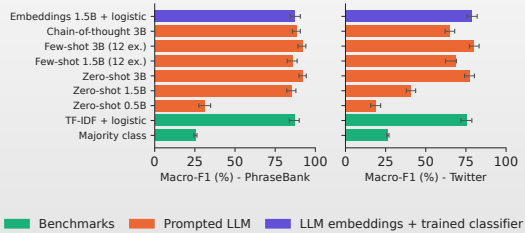
Data set	Labels	Train	Test	Neutral share (test)
Financial PhraseBank, 100 % agreement	positive, negative, neutral	1581	678	61.4 %
Twitter financial news	bullish, bearish, neutral	9 543	900	65.6 %

- PhraseBank: 4 838 sentences from news about Finnish listed companies, labelled by 5–8 annotators; we use the 2 259 with unanimous labels, split 70/30 stratified (seed 20261108).
- Twitter: published split; test = stratified sample of 900 of the 2 388 validation tweets.
- The examples for few-shot prompts are drawn from the training part only.

(Malo et al., 2014)

Prompted LLMs against a trained benchmark

- **In one sentence:** no single strategy wins everywhere; the trained classifiers use the labelled data, the prompts do not.



Source: Financial PhraseBank (Malo et al., 2014) and Twitter financial news (zeroshot). Data: test sets: 678 sentences (100 % agreement) and 900 tweets; 95 % bootstrap intervals.

- Best on PhraseBank: zero-shot 3B (92.2 %)
  - On Twitter: few-shot 3B (79.8 %)
  - Zero-shot 3B: 92.2 % and 77.4 %.

NN_L12_prompt_results

Accuracy and macro-F1 with 95 % bootstrap intervals

Method	Acc.	F1	95 % int.	Acc.	F1	95 % int.
	PhraseBank			Twitter		
majority class	61.4	25.4	[24.4; 26.3]	65.6	26.4	[25.7; 27.1]
TF-IDF + logistic	90.3	86.9	[83.8; 90.0]	81.4	75.5	[71.9; 78.7]
zero-shot 0.5B	62.8	31.0	[27.5; 34.8]	23.1	18.7	[15.7; 21.9]
zero-shot 1.5B	85.3	85.1	[82.2; 87.9]	37.8	40.9	[37.9; 43.7]
zero-shot 3B	93.1	92.2	[89.8; 94.4]	81.2	77.4	[74.2; 80.3]
few-shot 1.5B	87.8	85.9	[82.5; 88.7]	73.7	68.8	[62.3; 69.1]
few-shot 3B	93.4	92.1	[89.2; 94.1]	84.3	79.8	[77.2; 83.2]
chain-of-thought 3B	88.5	88.4	[85.7; 90.7]	64.7	65.1	[62.0; 68.0]
embeddings 1.5B + logistic	91.3	87.5	[84.5; 90.7]	83.0	79.0	[75.7; 82.0]

- Macro-F1 = mean of the three per-class F1 scores, so the small negative class counts as much as neutral; intervals from 1,000 bootstrap resamples of the test set; few-shot = mean of 3 seeds.
- Int. = 95 % bootstrap interval of the macro-F1.

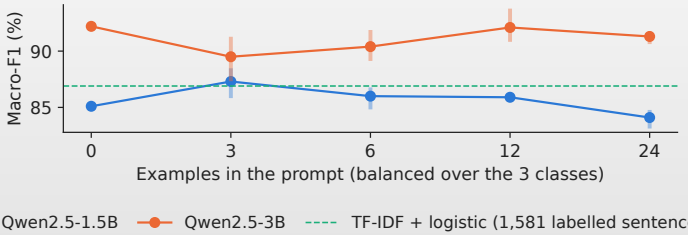
(own computation)

Experiment yourself: notebook §3

Effect of the number of examples in the prompt

- **In one sentence:** in-context learning: the examples change the output without changing a single weight.

$$\hat{c} = \arg\max_{c} \; p_\theta\big(c \mid (x_1, c_1), \ldots, (x_k, c_k), x\big)$$



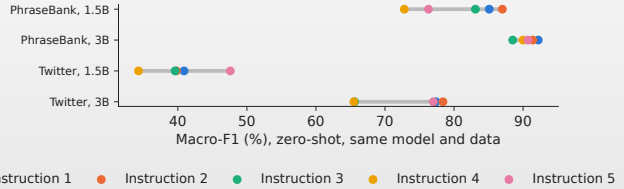
Source: Financial PhraseBank (Malo et al., 2014). Data: test set of 678 sentences; bars = range over 3 seeds.

- 1.5B: 85.1 % with no example, 84.1 % with 24
  - 3B: 92.2 % and 91.3 %
  - Seeds move 3B with 12 examples between 91.0 and 93.6 %.

NN_L12_fewshot_curve

Prompt sensitivity: effect of rewording the instruction

- **In one sentence:** five instructions with the same meaning, the same model and data: the spread is a source of model risk.



Source: Financial PhraseBank and Twitter financial news. Data: test sets as above; five paraphrases of the instruction.

- Range of macro-F1 over the five wordings: PhraseBank 3B 3.6 points, Twitter 3B 12.9 points, Twitter 1.5B 13.3 points.

- **In practice:** choose the prompt on a validation set and freeze it, like any hyper-parameter
  - Never on the test set.

NN_L12_prompt_sensitivity

Chain-of-thought: written reasoning before the answer

- **In one sentence:** generated reasoning tokens become extra context for the final label.

$$p(c \mid x) = \sum_{r} p(r \mid x)\, p(c \mid x, r) \;\approx\; p(c \mid x, \hat{r})$$

- **where**
  - r = reasoning text, $\hat{r}$ = the greedy one (at most two sentences, on average 35.9 words)
  - The label is parsed from "Label: . . . "
  - Parsed for 100.0 % of PhraseBank answers.

- **Macro-F1**: PhraseBank 88.4 % (zero-shot 3B 92.2 %)
  - Twitter 65.1 % (77.4 %).
- **Cost**: 389 ms per sentence against 48.5 ms for zero-shot.

- **In practice:** chain-of-thought helps on multi-step reasoning
  - For one-word labels on short text, test before paying for it.

(Wei et al., 2022)

Experiment yourself: notebook §5

An example of a chain-of-thought answer

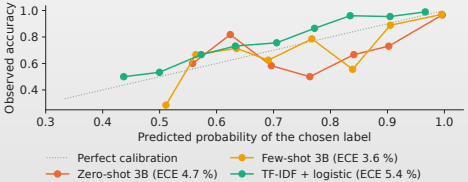
- **Input** (PhraseBank test set)
  - "CapMan said the deal 's effect on its cash flow for 2009 totals EUR3 .4 m , but the transaction would not affect its financial results for 2009 as it was executed at fair value ."
- **Output** of Qwen2.5-3B-Instruct, temperature 0
  - "Reasoning: The sentence indicates that the deal has a positive impact on CapMan's cash flow, which is generally seen as beneficial for investors. However, it also clarifies that the transaction does not affect their financial results, suggesting no immediate change to profitability. Given these points, the overall sentiment leans positive as it highlights a direct benefit. Label: positive"
- **How to read it**
  - The reasoning is a plausible story, not a faithful trace of the computation
  - Audit the label against the data, not the explanation against your intuition

(own computation)

Calibration of the label probabilities on real data

- **In one sentence:** a probability of 0.9 should be right nine times in ten; points below the diagonal mean over-confidence.

$$\mathrm{ECE} = \sum_{b=1}^{B} \frac{|S_b|}{n} \left|\mathrm{acc}(S_b) - \mathrm{conf}(S_b)\right|$$



Source: Financial PhraseBank (Malo et al., 2014). Data: test set of 678 sentences, 10 bins.

- Expected calibration error (ECE): zero-shot 3B 4.7 %, few-shot 3B 3.6 %, TF-IDF + logistic 5.4 %.

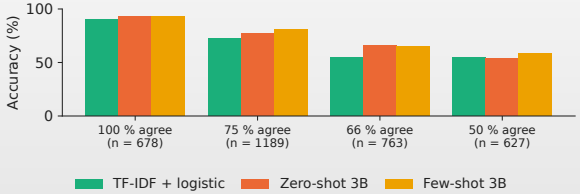
- **In practice:** recalibrate on a validation set (temperature scaling) before using LLM probabilities as risk scores.

(Guo et al., 2017)

NN_L12_calibration

Accuracy by level of annotator agreement

- **In one sentence:** sentences that split the annotators are genuinely ambiguous; no model reaches the unanimous-case accuracy on them.



Source: Financial PhraseBank (Malo et al., 2014). Data: sentences never used for training or for examples.

- Zero-shot 3B: 93.1 % at 100 % agreement, 53.7 % at 50 %
  - TF-IDF: 90.3 % and 54.9 %.

- **In practice:** report the label quality of your test set
  - A model cannot be more certain than the people who labelled it.

NN_L12_agreement

Confusion matrix of the best prompt

True \ predicted	positive	negative	neutral
positive	156	2	13
negative	1	82	8
neutral	20	3	393

- **Model**: zero-shot 3B, PhraseBank test set.
- **Neutral read as positive**: 20 sentences
  - **positive read as neutral**: 13.
- **Negative and positive** are rarely swapped: 1 + 2.

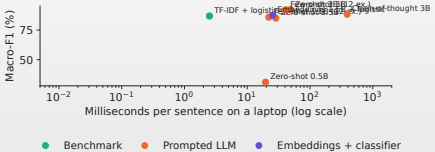
- Errors sit on the neutral border: “profit rose but less than expected” is the hard case, not good versus bad news.

(own computation)

Experiment yourself: notebook §6

Cost against accuracy: milliseconds per sentence on a laptop

- **In one sentence:** the cheapest model is thousands of times faster than the slowest; the accuracy gap is small.



Source: Financial PhraseBank (Malo et al., 2014). Data: test set of 678 sentences; time measured on a laptop graphics processor.

- TF-IDF: 2.50 ms per sentence
  - Zero-shot 3B: 48.5 ms
  - Chain-of-thought 3B: 389 ms.

- **In practice:** with labels, train a small model
  - Without labels, prompt an LLM
  - For millions of documents, the cost decides.

NN_L12_cost_accuracy

LLM embeddings with a small classifier as a low-cost alternative to fine-tuning

- **In one sentence:** freeze the language model, average its last hidden states, and fit a logistic regression on top.

$$\mathbf{e}(x) = \frac{1}{|x|} \sum_{t=1}^{|x|} \mathbf{h}_t^{(L)}, \qquad \hat{p}(c \mid x) = \operatorname{softmax}\left(\mathbf{W}\mathbf{e}(x) + \mathbf{b}\right)_c$$

- **Macro-F1**: PhraseBank 87.5 %, Twitter 79.0 %.
- **Uses the labels** like TF-IDF, but with the language knowledge of the LLM.

- **where**
  - $\mathbf{h}_t^{(L)} \in \mathbb{R}^{1536}$ = last hidden state of Qwen2.5-1.5B
  - $\mathbf{W}$ fitted with an L2 penalty chosen by 5-fold cross-validation.

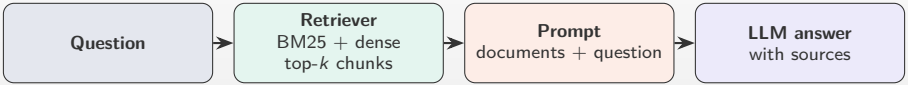
- **In practice:** before full fine-tuning, try a probe: minutes on a laptop, no gradient through the LLM.

Experiment yourself: notebook §6

Outline

- Next-token prediction and large language models
- Pre-training, instruction tuning and alignment
- Evaluation of prompt engineering on labelled data
- Retrieval-augmented generation on ECB speeches
- Language models as risk models: LLM-VaR
- Risks and governance

RAG: document retrieval followed by answer generation



$$p(y \mid q) \approx \sum_{z \in \mathrm{top}_k(q)} p_\eta(z \mid q)\, p_\theta(y \mid q, z)$$

- **Retrieval-augmented generation (RAG)**: z = retrieved chunk, p_η = retriever, p_θ = generator.
- **Our corpus**: 400 ECB speeches, 8 400 chunks of 200 words (overlap 50).

(Lewis et al., 2020)

Three retrievers: shared words, weighted terms, dense embeddings

- **In one sentence:** lexical retrievers match words, dense ones match meaning, a hybrid combines both; the query here is a speech title.

$$\mathrm{BM25}(q,z) = \sum_{w \in q} \mathrm{idf}(w) \frac{f(w,z)(k_1+1)}{f(w,z) + k_1\left(1 - b + b\frac{|z|}{|z|}\right)}$$

$$s_{\text{dense}} = \cos\Big(\mathbf{e}(q), \mathbf{e}(z)\Big), \quad s_{\text{RRF}}(z) = \sum_m \frac{1}{60 + \mathrm{rank}_m(z)}$$

- **Mean reciprocal rank (MRR)**: BM25 0.515, dense 0.567, hybrid 0.622.

- **where**
  - $f(w,z)$ = count of word w in chunk z
  - $k_1 = 1.5$, $b = 0.75$
  - Best Matching 25 (BM25)
  - $\mathbf{e}$ = bge-small embedding
  - RRF = rank fusion.

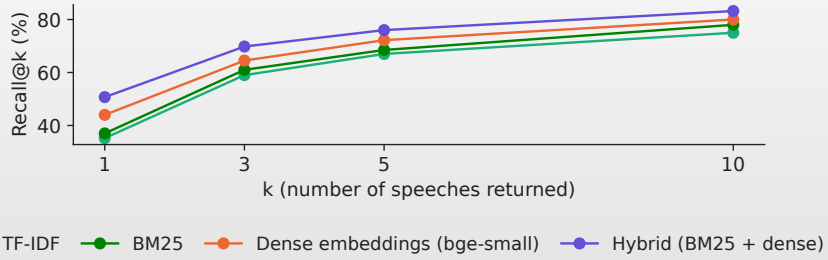
- **In practice:** the answer cannot be better than what is retrieved: evaluate the retriever on its own first.

(Robertson & Zaragoza, 2009; Xiao et al., 2023)

Experiment yourself: notebook §7

Retrieval of the correct speech from its title

- **In one sentence:** recall@k = share of the 400 titles whose speech is among the top k results.



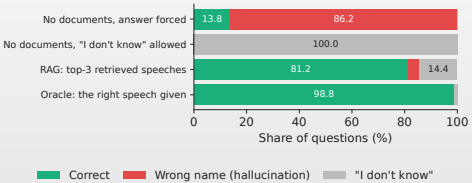
Source: European Central Bank, speeches dataset. Data: 400 speeches, 8400 chunks; query = speech title.

- Recall@1: TF-IDF 35.2 %, BM25 37.0 %, dense 44.0 %, hybrid 50.7 %
  - Recall@10 hybrid 83.2 %.

NN_L12_rag_recall

Answers with and without retrieved documents

- **In one sentence:** question: which board member gave the speech with this title? 160 speeches since 15 Jan 2025, after the model's training data.



Source: European Central Bank, speeches dataset. Data: 160 single-speaker speeches from 15 Jan 2025 to 30 Oct 2026; model Qwen2.5-3B-Instruct.

- Forced answer without documents: 86.2 % wrong names, Philip Lane 137 times
  - RAG 81.2 % correct, oracle 98.8 %
  - The most frequent speaker alone: 35.6 %.

- **In practice:** a confident wrong name is the costly failure: require sources in every answer and check them.

NN_L12_rag_qa

Design choices for a RAG system in a bank

- **Chunking**
  - 200 words with 50 overlap here; tables and footnotes need their own rules
- **Metadata**
  - Speaker, date and title travel with each chunk; the answer cites them
- **Retriever**
  - Hybrid BM25 + dense: recall@3 = 73.1 % on the question-answering speeches
- **Abstention**
  - Allow “I don’t know”: without documents the model then abstains on 100.0 % of questions
- **Evaluation**
  - A fixed set of questions with known answers, rerun after every change

Outline

- Next-token prediction and large language models
- Pre-training, instruction tuning and alignment
- Evaluation of prompt engineering on labelled data
- Retrieval-augmented generation on ECB speeches
- Language models as risk models: LLM-VaR
- Risks and governance

LLM-VaR: a quantile forecast from the return history in the prompt

- **In one sentence:** no estimation and no fine-tuning: the model reads 250 returns and writes the Value-at-Risk (VaR).

$$\widehat{\mathrm{VaR}}^{\alpha}_{t+1} = \mathrm{LLM}\big(\mathrm{prompt}(r_{t-249}, \ldots, r_t; \alpha)\big)$$

$$\mathrm{LR}_{uc} = -2 \ln \frac{(1-\alpha)^{n_0} \alpha^{n_1}}{(1-\hat{\pi})^{n_0} \hat{\pi}^{n_1}} \sim \chi^2_1$$

- **where**
  - r_t = daily log return of the S&P 500 (%)
  - $\alpha \in \{1\,\%, 5\,\%\}$
  - Temperature 0
  - n_1 = violations
  - Kupiec likelihood ratio (LR) test.

- **Our replication**: Qwen2.5-3B-Instruct, 459 days from 2 Jan 2025 to 30 Oct 2026.
- **Benchmarks**: historical simulation (HS), 250 days
  - Generalised autoregressive conditional heteroskedasticity (GARCH), GARCH(1,1)-t.

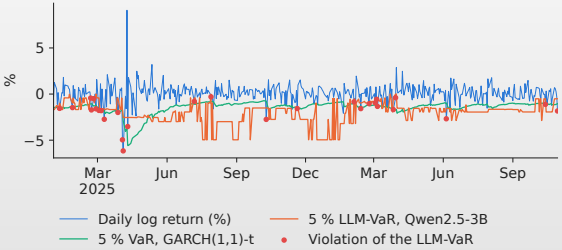
- **In practice:** a 3B open model on a laptop is the cheapest test of the idea
  - The paper used large commercial models.

(Pele et al., 2026; Kupiec, 1995)

Experiment yourself: notebook §9

LLM-VaR with a 3B open model on the S&P 500

- **In one sentence:** the language model's 5 % VaR against GARCH; red dots mark the days the LLM-VaR was breached.



Source: Yahoo Finance, daily quotes. Data: 2 Jan 2025 to 30 Oct 2026, S&P 500, trading days.

- 5 %: LLM 28 violations, GARCH 35, HS 23, expected 23.0.
- Mean 5 % VaR: LLM −1.99 %, GARCH −1.42 %
  - Correlation of the two 0.05: the small model covers by being cautious, not by tracking volatility.

NN_L12_llm_var

Backtest of LLM-VaR against the classical benchmarks

Model	Viol. 1 %	Rate (%)	p_{uc}	Viol. 5 %	Rate (%)	p_{uc}	Mean VaR 5 %
LLM-VaR (Qwen2.5-3B)	4	0.87	0.777	28	6.10	0.295	−1.99
HS (250 days)	6	1.31	0.528	23	5.01	0.991	−1.56
GARCH(1,1)-t	9	1.96	0.067	35	7.63	0.016	−1.42

- Expected violations: 4.6 at 1 % and 23.0 at 5 %; green p-value: coverage not rejected at 10 %.
- Answers that could not be read as a number: 100.0 % parsed at 5 %, 100.0 % at 1 %; an unreadable answer keeps the previous day's value.
- Model size matters: the published study evaluated much larger models; results here describe a 3B model only.

(own computation; Pele et al., 2026)

Experiment yourself: notebook §9

Outline

- Next-token prediction and large language models
- Pre-training, instruction tuning and alignment
- Evaluation of prompt engineering on labelled data
- Retrieval-augmented generation on ECB speeches
- Language models as risk models: LLM-VaR
- Risks and governance

Six risks of LLMs in a financial institution, and the test for each

Risk	Seen in this lecture	Control
Hallucination	86.2 % wrong speaker names without documents	RAG with sources; abstention allowed
Leakage / contamination	lower perplexity on older speeches	test on text after the cut-off
Prompt sensitivity	12.9 macro-F1 points across wordings	select on validation, freeze, version
Miscalibration	ECE 4.7 % zero-shot, 5.4 % TF-IDF	temperature scaling, thresholds on validation
Cost and latency	389 ms vs 2.50 ms per sentence	smallest model that passes the test
Data protection	client documents in prompts	local models; duties of the European Union (EU) Artificial Intelligence (AI) Act for high-risk uses such as credit scoring

⊡ The EU AI Act is Regulation (EU) 2024/1689.

Quiz 1: temperature

- You classify 10,000 news items with an LLM and get different labels on two runs of the same item
  - What do you change first?

Answer

- Set temperature to 0 (or read label probabilities): sampling at $T > 0$ is random by construction.

Quiz 2: evaluation

- A vendor reports 95 % accuracy of its LLM on Financial PhraseBank
  - What two questions do you ask?

Answer

- Which agreement subset (the unanimous one is easiest) and was PhraseBank in the training data (published 2014, widely copied)? Ask for macro-F1 too: 61.4 % of the sentences are neutral.

Quiz 3: RAG

- Your RAG assistant answers wrongly although the right document exists in the corpus
  - Where do you look first?

Answer

- At the retriever: check whether the document is in the top k (recall@k)
  - Only then at the prompt and the model.

Wrap-up

- An LLM is a next-token model
  - Perplexity, temperature and tokens are its basic quantities.
- A prompt is a model: score it on held-out labels, with intervals, seeds and paraphrases.
- With labels, small trained models and LLM embeddings are hard to beat
  - Without labels, prompting gives a strong start.
- RAG turns hallucinated answers into sourced ones
  - Evaluate the retriever first.
- Next: time-series foundation models for energy and finance (Lecture 13).

Research from our group

- **LLM-VaR and, for the expected shortfall (ES), LLM-ES**: general-purpose LLMs estimate both risk measures zero-shot from a structured prompt with past returns
  - Backtested against classical risk models (Pele, Bolovăneanu, Lin, Ren, Ginavar, Spilak, Andrei, Toma, Lessmann & Härdle, 2026, Expert Systems with Applications).
- **LLM-driven stock prediction**: LLaMA 3.3 combines price history with news sentiment in the prompt and outperforms LLaMA 3.1 and an autoregressive integrated moving-average benchmark (Tak & Pele, 2025, PICBE).
- **Central-bank narratives**: a transformer-based sentiment index of the National Bank of Romania's minutes improves Romanian inflation forecasts (Amza, Bolovăneanu & Pele, 2026, PICBE) — the same kind of text as our ECB speeches.
- Findings as stated in the papers' abstracts; numbers on the other slides come from our own data.

(Bucharest University of Economic Studies, ASE; Romanian Academy, Institute for Economic Forecasting, IPE)

Explore it visually

- **Transformer Explainer**: the temperature of this lecture on a live GPT-2: at low temperature one continuation takes almost all the probability, at high temperature unlikely tokens become plausible.
- **Mixture of experts (M. Grootendorst)**: a router sends each token to a few experts, so a model can have many parameters while using only some of them per token.
- **LLM Inference Handbook**: what happens between the prompt and the answer: the prefill of the prompt, token-by-token decoding, the cache of keys and values, and the time to the first token.
- **LLM Visualization (B. Bycroft)**: the same architecture at several sizes, from a small GPT to GPT-2 and GPT-3.

Further reading and resources

- **Hugging Face** course on LLMs: huggingface.co/learn/llm-course.
- **Qwen2.5** models and report: huggingface.co/Qwen.
- **Prompting guide**: promptingguide.ai.
- **ECB speeches** data set: ecb.europa.eu.
- **Quantinar**: The Next Word Problem: quantinar.com.
- **This lecture**: notebook, seminar and Quantlets — github.com/danpele/NEURAL_BIZ.

References (1/2)

- Amza, A., Bolovăneanu, V. & Pele, D. T. (2026). Inflation forecasting with monetary policy narratives: Evidence from an emerging economy. *Proceedings of the International Conference on Business Excellence*, 20(1), 75–83. doi:10.2478/picbe-2026-0012
- Brown, T. B. et al. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33. arXiv:2005.14165
- Guo, C., Pleiss, G., Sun, Y. & Weinberger, K. Q. (2017). On calibration of modern neural networks. *Proceedings of the 34th International Conference on Machine Learning*, 1321–1330. arXiv:1706.04599
- Hoffmann, J. et al. (2022). Training compute-optimal large language models. *Advances in Neural Information Processing Systems*, 35. arXiv:2203.15556
- Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L. & Chen, W. (2022). LoRA: Low-rank adaptation of large language models. *International Conference on Learning Representations*. arXiv:2106.09685
- Kaplan, J. et al. (2020). Scaling laws for neural language models. arXiv:2001.08361
- Kupiec, P. H. (1995). Techniques for verifying the accuracy of risk measurement models. *The Journal of Derivatives*, 3(2), 73–84. doi:10.3905/jod.1995.407942
- Lewis, P. et al. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems*, 33. arXiv:2005.11401
- Malo, P., Sinha, A., Korhonen, P., Wallenius, J. & Takala, P. (2014). Good debt or bad debt: Detecting semantic orientations in economic texts. *Journal of the Association for Information Science and Technology*, 65(4), 782–796. doi:10.1002/asi.23062
- Ouyang, L. et al. (2022). Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems*, 35. arXiv:2203.02155

References (2/2)

- Pele, D. T., Bolovăneanu, V., Lin, M.-B., Ren, R., Ginavar, A. T., Spilak, B., Andrei, A.-V., Toma, F.-M., Lessmann, S. & Härdle, W. K. (2026). In the beginning was the Word: LLM-VaR and LLM-ES. *Expert Systems with Applications*, 295, 128676. doi:10.1016/j.eswa.2025.128676
- Qwen Team (2024). Qwen2.5 technical report. arXiv:2412.15115
- Rafailov, R., Sharma, A., Mitchell, E., Ermon, S., Manning, C. D. & Finn, C. (2023). Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36. arXiv:2305.18290
- Robertson, S. & Zaragoza, H. (2009). The probabilistic relevance framework: BM25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4), 333–389. doi:10.1561/1500000019
- Sennrich, R., Haddow, B. & Birch, A. (2016). Neural machine translation of rare words with subword units. *Proceedings of the 54th Annual Meeting of the ACL*, 1715–1725. doi:10.18653/v1/P16-1162
- Tak, R. & Pele, D. T. (2025). LLM-driven stock prediction: Capturing market trends with LLaMA. *Proceedings of the International Conference on Business Excellence*, 19(1), 529–540. doi:10.2478/picbe-2025-0043
- Wei, J. et al. (2022). Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35. arXiv:2201.11903
- Xiao, S., Liu, Z., Zhang, P. & Muennighoff, N. (2023). C-Pack: Packaged resources to advance general Chinese embedding (BAAI general embedding). arXiv:2309.07597
- Zhao, Z., Wallace, E., Feng, S., Klein, D. & Singh, S. (2021). Calibrate before use: Improving few-shot performance of language models. *Proceedings of the 38th International Conference on Machine Learning*, 12697–12706. arXiv:2102.09690

Appendix A: the prompt used for zero-shot classification

- **System**
  - "You are a careful financial analyst."
- **User**
  - "Classify the sentiment of this sentence from financial news, from the point of view of an investor in the company. Answer with one word: positive, negative or neutral. Sentence: ... Answer:"
- **Few-shot**
  - The same, with "Sentence: ... Answer: label" pairs before the input; examples balanced across classes and shuffled
- **Chain-of-thought**
  - "... First explain your reasoning in at most two short sentences, then give the final answer on a new line as 'Label: <positive|negative|neutral>'."

Appendix B: implementation details

- Models in float16 on the laptop's graphics processor (PyTorch)
  - Hugging Face transformers
  - Scikit-learn for TF-IDF, logistic regression and bootstrap.
- Seeds: 20261108 for splits and samples
  - Few-shot examples drawn with seeds 20261108, $+1$ and $+2$.
- The whole analysis took 2.2 minutes on its last run (model outputs cached)
  - First run about 40 minutes.
- Per-item predictions are published with the Quantlets, so every number can be recomputed without a graphics card.