

Processamento de Linguagem Natural e Processamento de Imagens

Aula 4

James Moraes de Almeida



Objetivos desta aula

- Ao final, você deve saber:
 - Explicar como o motor de regex procura um padrão num texto.
 - Ler e escrever os blocos da linguagem: classes, quantificadores, âncoras.
 - Reconhecer as três formas de uma extração falhar — inclusive a silenciosa.
 - Montar um pipeline de extração sobre um corpus real de resumos.
 - Medir a qualidade do que foi extraído, em vez de confiar na impressão.
 - Identificar onde a regra por padrão desiste, e por quê.

De 6.228 resumos a uma tabela

- Um resumo da Web of Science diz:
 - "The yield strength of the AlCuNiFeCr alloy reached 1511 MPa at room temperature."
- A linha de tabela que queremos:
 - composição = AlCuNiFeCr · yield strength = 1511 MPa
- Agora multiplique por 6.228 resumos.
 - Ninguém lê isso à mão. E é para esse tipo de trabalho que o regex existe.

O corpus desta aula

- Exportações da Web of Science sobre ligas de alta entropia.
 - 6.228 resumos com texto.
 - 48.719 sentenças.
 - 754 resumos mencionam "yield strength".
- A pergunta da aula não é "dá para achar?".
 - É "quanto dá para achar, e com quanta confiança?".

Cinco datas

- 1951 — Stephen Kleene descreve os "eventos regulares".
- 1968 — Ken Thompson publica um algoritmo de busca; dele nasce o grep.
- 1986 — Henry Spencer publica na Usenet uma biblioteca de regex em C.
- 1988 — o Perl 2 adota essa biblioteca e leva os regexes muito além.
- hoje — "compatível com Perl" (PCRE) é o padrão de fato.
 - grep -P, regex101, editores de texto, Python, Java, JavaScript.

Como o motor procura

The yield strength of the alloy.



padrão: [Yy] i e l d

Os blocos da linguagem

- Classes — o que pode ocupar uma posição:
 - `[Yy]` uma das duas · `[A-Z]` qualquer maiúscula · `[^abc]` qualquer um menos esses
 - `\d` dígito · `\s` espaço · `\w` letra, dígito ou `_` · `.` qualquer caractere
 - `\.` ponto literal — a barra invertida TIRA o poder do metacaractere
 - é a diferença entre `.+` (qualquer coisa) e `\. \d+` (o ".1" de A10.1)
- Quantificadores — quantas vezes:
 - `+` uma ou mais · `*` zero ou mais · `?` zero ou uma · `{4,8}` de quatro a oito
- Âncoras — onde, sem consumir nada:
 - `\b` fronteira de palavra · `^` início · `$` fim
- Grupos e alternância:
 - `(...)` captura · `(?:...)` agrupa sem capturar · `a|b` um ou outro

Grupos: capturar ou só agrupar

- Os parênteses fazem duas coisas ao mesmo tempo: agrupam e capturam.
- Agrupar — aplicar um quantificador a um pedaço inteiro:
 - `(ab)+` repete o "ab"; sem os parênteses, `ab+` repetiria só o "b".
- Capturar — guardar o que casou para usar depois. Mas há um efeito colateral:
 - com grupo, o `findall` devolve o GRUPO, não o casamento inteiro.
 - `re.findall(r"(no ){1,3}", "no no no") → ['no ']`
 - `re.findall(r"(?:no ){1,3}", "no no no") → ['no no no ']`
- `(?:...)` agrupa sem capturar — é o que você quer quando só precisa repetir.
 - No regex da composição, sem o `(?:...)` o `findall` devolve `('.1', 'Cr0.1', '.1')`
 - em vez de `'Al0.1Cu0.3Ni0.2Fe0.3Cr0.1'`. O grupo rouba a resposta.

`\b` : uma posição, não um caractere

Meus átomos preferidos são Fe, C e Br.

`[A-Z][a-z]?`

A fronteira fica ENTRE as colunas — e não ocupa nenhuma.

Guloso e preguiçoso

1511	MPa e 2000 MPa a 25 C
------	-----------------------

\d+.+MPa

devolvendo caractere por caractere, e testando MPa a cada passo...

Da letra à liga, em camadas

- Um símbolo químico: maiúscula, minúscula opcional.
 - $[A-Z][a-z]? \rightarrow \text{Al, C, Fe}$
- Com o índice estequiométrico:
 - $[A-Z][a-z]?\d*(?:\.\d+)? \rightarrow \text{Al}_{0.1}, \text{Ni}_2$
- Repetido de quatro a oito vezes, além do primeiro:
 - $[A-Z][a-z]?\d*(?:\.\d+)?(?:[A-Z][a-z]?\d*(?:\.\d+)?){4,8}$
 - $\rightarrow \text{Al}_{0.1}\text{Cu}_{0.3}\text{Ni}_{0.2}\text{Fe}_{0.3}\text{Cr}_{0.1}$ (5 elementos = 1 + 4)
 - O {4,8} não é sintaxe — é a decisão de só aceitar ligas de 5 a 9 elementos.
 - CoCrNi não casa. Escreva isso no comentário, ou em seis meses nem você lembra por quê.
- Em re.VERBOSE o padrão se explica sozinho, e roda exatamente igual.

O regex da composição, termo a termo

Al_{0.1}Cu_{0.3}Ni_{0.2}Fe_{0.3}Cr_{0.1}

`[A-Z][a-z]? \d* (?:\.\d+)?`

um bloco = um elemento com o seu índice

O erro que não avisa

microhardness of 5.14 GPa

\d+ \s [TGMMkk] P a

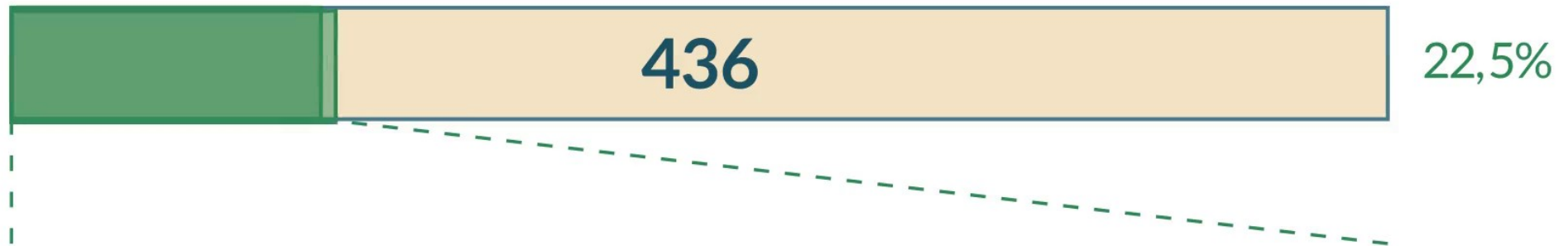
o motor avança e tenta de novo...

Três formas de falhar

- Não achar nada — a lista vem vazia.
 - Visível. Regex estreito demais, ou o texto usa outra grafia.
- Achar demais — vários valores para a mesma composição.
 - Visível. É o que acontece ao trocar search por findall.
- Achar errado — um número plausível, e errado.
 - Invisível. Contamina a planilha sem deixar rastro.
 - Só aparece se alguém conferir uma amostra à mão.

O que sobra de 6.228 abstracts

...e um valor em Pa



Onde a regra desiste

- Duas extrações que a revisão manual precisou remover:
 - "...yield strength of V35Ti35Fe15Cr10Zr5 was about 500 MPa higher than..."
 - o número é uma diferença, não o valor da propriedade.
 - "...maximum compressive yield strength of 1479 MPa"
 - é o limite em compressão — outra grandeza.
- Nos dois casos a sintaxe está certa e o sentido está errado.
 - Nenhum regex resolve: falta entender a frase. É a próxima parte da aula.

Fim da Parte 1 — Expressões regulares

- O que fica desta parte:
 - O motor tenta da esquerda para a direita e para no primeiro casamento inteiro.
 - Os quantificadores são gulosos por padrão — e isso junta dois valores num só.
 - A âncora `\b` não consome caractere: só pergunta se ali muda de palavra.
 - Três formas de falhar: não achar, achar demais e achar errado.
- O placar honesto: 45 linhas de 48.719 sentenças, 44 delas corretas.
 - Regex casa forma. Não entende o sentido da frase.

Mão na massa

- Abra o Regex101.ipynb.
- Parte 1 — do regex numa sentença ao pipeline nos 6.228 resumos.
 - Três versões do código, cada uma consertando um erro da anterior.

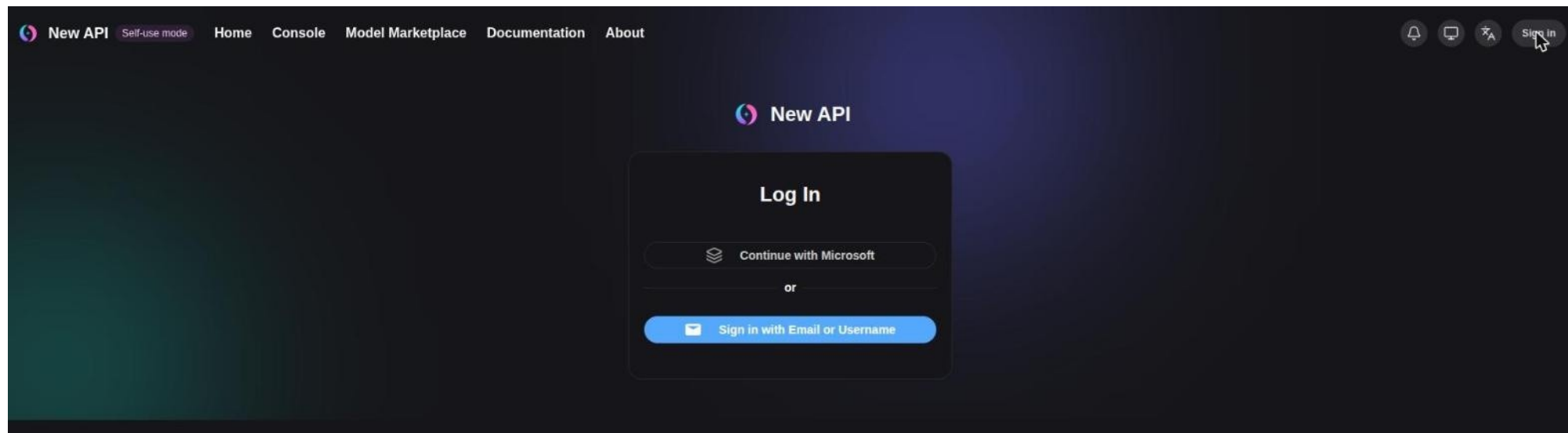
Parte 2 — Extração com LLM

- A mesma tarefa, nas mesmas frases, agora com um modelo de linguagem.
- O caminho:
 - Acesso à IlumA e o seu token pessoal.
 - System, user e assistant prompt.
 - Temperatura, top-p e top-k — por alto.
 - O extrator em JSON, e a comparação lado a lado com o regex.
- A pergunta que fica: o que o modelo resolve — e o que ele quebra?

IlumA — o gateway de LLM da Ilum

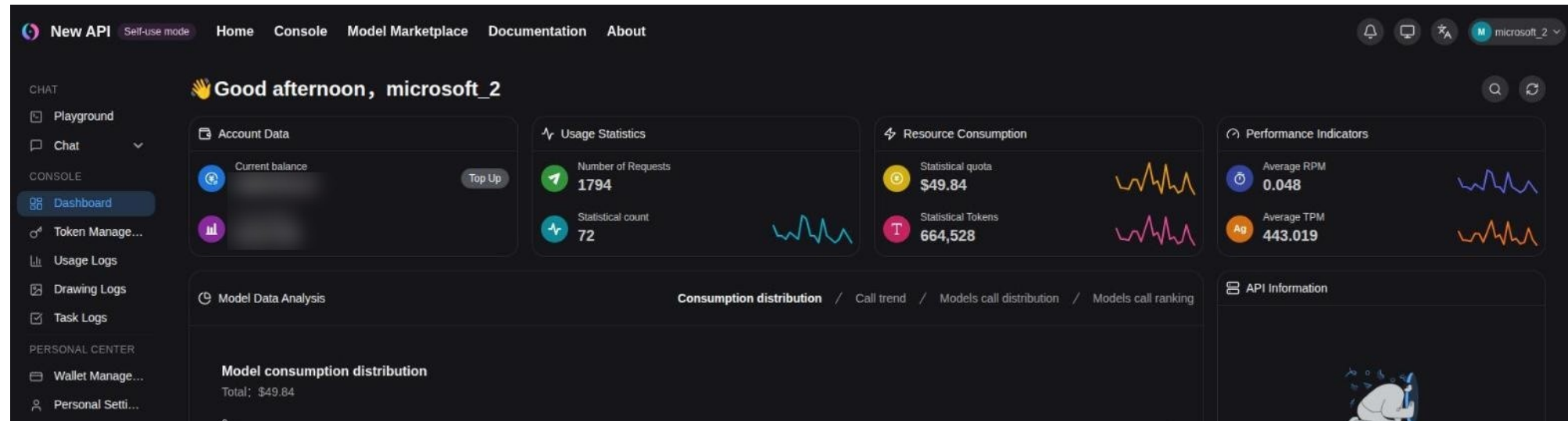
- Um endereço único que fala o mesmo protocolo da API da OpenAI.
 - `base_url = https://iluma.cnpem.br:4000/v1`
 - `modelo = ilum`
- Em relação a qualquer exemplo da OpenAI, muda só a `base_url`.
- Você precisa de um token pessoal — e cada chamada gasta cota da sua conta.

Passo 1 — entre com a conta da Ilum



iluma.cnpem.br:4000 → Sign in → Continue with Microsoft

Passo 2 – Console → Token Management



O Console é o painel da sua conta; o menu da esquerda tem os tokens.

Passo 3 – Create token

Create New Token

Basic Information
Set token basic information

Name *

meu-token-aula04

Token Group

Token group, default is your group

Expiration time *

Expiration time quick settings

1969-12-31 20:59:59

Never expires

A month

One day

One hour

New quantity *

1

When creating in batches, a random suffix will be automatically added to the name

Quota Settings
Set token available quota and quantity

Amount

\$ 0.000000

Use raw quota input

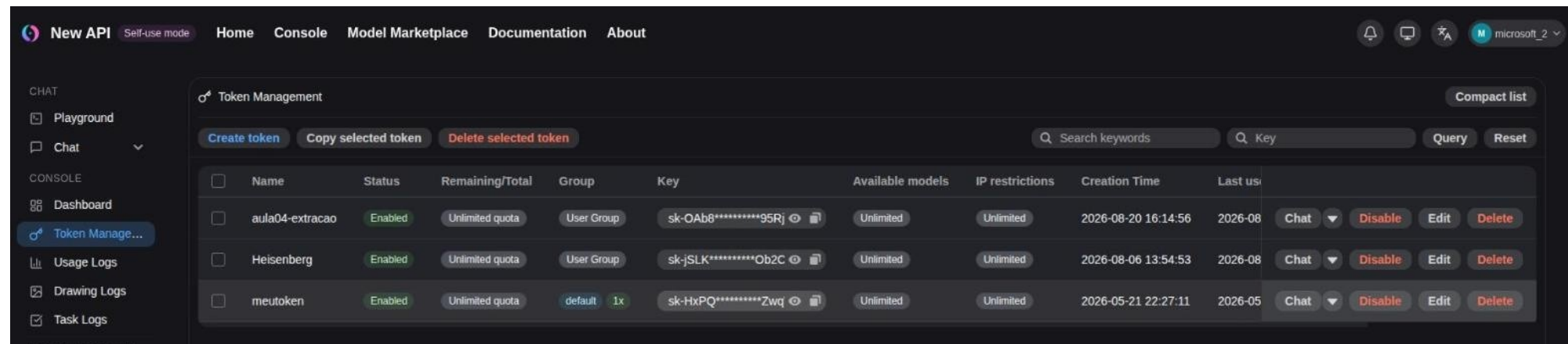
Unlimited quota

The quota of the token is only used to limit the maximum quota usage of the token itself, and the actual usage is limited by the remaining quota of the account

Submit Cancel

Um nome que você reconheça, o grupo padrão, e Submit.

Passo 4 — copie a chave



The screenshot shows the 'Token Management' page in the Azure OpenAI Studio. The interface includes a sidebar with navigation options like 'New API', 'Self-use mode', 'Home', 'Console', 'Model Marketplace', 'Documentation', and 'About'. The main area displays a table of API keys with columns for Name, Status, Remaining/Total, Group, Key, Available models, IP restrictions, Creation Time, and Last used. Three keys are listed: 'aula04-extracao', 'Heisenberg', and 'meutoken'. Each key has a 'Copy' icon and a 'Delete' button. The 'meutoken' key is highlighted with a green background.

Name	Status	Remaining/Total	Group	Key	Available models	IP restrictions	Creation Time	Last used
aula04-extracao	Enabled	Unlimited quota	User Group	sk-OAb8*****95Rj	Unlimited	Unlimited	2026-08-20 16:14:56	2026-08
Heisenberg	Enabled	Unlimited quota	User Group	sk-jSLK*****Ob2C	Unlimited	Unlimited	2026-08-06 13:54:53	2026-08
meutoken	Enabled	Unlimited quota	default 1x	sk-HxPQ*****Zwq	Unlimited	Unlimited	2026-05-21 22:27:11	2026-05

O token é uma senha: não vai para o notebook, nem para o GitHub.

Os três papéis de uma conversa

- A conversa é uma lista de mensagens, e cada uma tem um papel.
- system — quem o modelo é e que regras ele segue.
 - Vale para a conversa inteira: escrito uma vez, usado em milhares de chamadas.
- user — o pedido de agora.
 - No nosso caso, a sentença de onde queremos extrair os dados.
- assistant — o que o modelo respondeu.
 - A resposta volta para a lista, e é dali que a conversa continua.

O prompt de assistente

- O modelo não tem memória. Cada chamada é independente da anterior.
 - Ele não sabe o que você perguntou há dez segundos.
- O que parece lembrança é você reenviando a lista inteira, toda vez.
 - As respostas anteriores voltam como mensagens assistant.
 - A memória não está no modelo: está na lista que você carrega.
- E nada obriga a mensagem assistant a ter acontecido de verdade.
 - Um par (user, assistant) escrito por você vira um exemplo do formato.
 - Isso é few-shot: mostrar funciona melhor — e é mais curto — que descrever.
 - Vale sobretudo quando a saída tem forma rígida, como o nosso JSON.

Temperatura, top-p e top-k

- O modelo não escolhe a próxima palavra: ele sorteia a partir de uma distribuição.
- temperature — o quanto o sorteio arrisca.
 - Perto de 0, quase sempre a palavra mais provável. Acima de 1, ele ousa.
- top-p — só as mais prováveis até a soma chegar a p. Corta a cauda.
- top-k — a mesma ideia, por contagem fixa: só as k mais prováveis.
- O manual manda usar temperature = 0 para extração. Aqui não dá.
 - A IlumA roda um Qwen: abaixo de ~0,5 a chamada trava — sem erro e sem retorno.
 - Nosso piso é 0,5. Manual e instalação nem sempre coincidem: descobre-se testando.
 - Os três voltam com calma nas próximas aulas.

A mesma extração, agora com o modelo

- O system prompt não descreve padrões de texto.
 - Descreve o que conta como um dado válido — em português.
- As duas regras que salvam os casos que o regex errou:
 - "um número que aparece como diferença não é um valor"
 - "se o ensaio for de compressão, registre a condição"
- A resposta sai em JSON e entra direto num DataFrame.

Regex ou LLM?

- Regex — para o que tem forma fixa: unidades, DOIs, códigos, datas.
 - Rápido, de graça, determinístico. E nunca inventa um número.
 - Mas não entende a frase.
- LLM — para o que depende do sentido.
 - Entende a diferença, a compressão, o "respectively".
 - Mas custa, demora, e pode alucinar.
- **Quase todo pipeline sério usa os dois.**

Mão na massa

- Abra o Regex101.ipynb.
- Parte 2 — a mesma extração com a IlumA.
 - Tutorial do token, System/User prompt e a comparação lado a lado.