Knowledge Discovery of Emerging Developer Skill Profiles in Telegram Community Using Semantic Embeddings

Matheus L. Braga[1], Pedro L. P. da Silva[2]

[1] Instituto Federal Catarinense, Brasil
matheus.braga@ifc.edu.br
[2] Universidade do Estado de Santa Catarina, Brasil
pedrolpompeu@gmail.com

Abstract. This article proposes a pipeline to discover professional profiles of software developers from job posts published on Telegram. The corpus consists of 22,243 messages collected between May 2021 and January 2026 from a public Brazilian group dedicated to IT vacancies; a two-stage filter combining deterministic rules and semantic validation reduced the set to 7,264 valid posts. A taxonomy of 108 technical and contractual competences was applied to the corpus, three embedding models were compared, and the reduced space was partitioned with K-Means. The number of clusters followed internal indices restricted by admissibility criteria defined beforehand and decided by an independent evaluator, resulting in 23 profiles with mean topic coherence of 0.522. Labelling and auditing relied on models from different families, so that no model judged its own output. This choice mattered: the cross-architecture protocol spread the scores instead of concentrating them, and showed that about a fifth of the corpus forms clusters built around posting format rather than occupational content. With each competence expressed as a share of monthly posts and multiple-testing correction applied, only three of 50 series decline significantly and none rises.

CCS Concepts: • **Computing methodologies** → *Natural language processing*; *Unsupervised learning*; • **Information systems** → *Data mining*.

Keywords: Knowledge Discovery, Labour Market Intelligence, Semantic Embeddings, Topic Modeling, Unsupervised Clustering

1. INTRODUÇÃO

O mercado de trabalho em tecnologia da informação muda rápido. Novas linguagens, frameworks e especializações aparecem o tempo todo, e isso dificulta o acompanhamento por métodos tradicionais de análise [Siswipraptini et al. 2023]. O campo de *Labour Market Intelligence* (LMI) surgiu para lidar com esse problema, propondo formas computacionais de extrair e interpretar tais movimentos a partir de fontes digitais [Boselli et al. 2017; Colace et al. 2019], inclusive para medir a distância entre o que as instituições de ensino formam e o que as empresas procuram [Alibasic et al. 2022]. Portais como LinkedIn e Glassdoor cobrem bem parte desse cenário, mas grupos de mensageria ocupam espaço como canal de divulgação. A diferença não está no tamanho da audiência, e sim no formato: o recrutador publica direto, em texto corrido, sem campos padronizados nem revisão editorial. Ganha-se velocidade e perde-se estrutura, o que impede o uso direto de taxonomias fixas.

Ainda assim, esse material pode mostrar o que as empresas buscam no dia a dia [Colace et al. 2019]. O obstáculo é que ele circula sem padronização, o que limita os métodos estatísticos usuais [Sun et al. 2022] e exige automatização para dar conta de volumes crescentes [Doğan et al. 2022]. Vários caminhos já foram testados. Algoritmos como o K-Means revelam exigências latentes em grandes conjuntos de anúncios [Debao et al. 2021]; abordagens hierárquicas montam taxonomias de competências técnicas [Siswipraptini et al. 2023]; o LDA (*Latent Dirichlet Allocation*) acompanha como os temas das vagas

evoluem no tempo [Colace et al. 2019; Wibowo 2025]; e redes de habilidades apontam o que está em alta ou em queda [Sibarani and Scerri 2020]. Análises assim permitem antecipar tendências antes que elas cheguem a registros formais [Ozcan et al. 2020].

Trabalhos recentes ampliaram o uso de embeddings e de métodos não supervisionados sobre corpora de mercado de trabalho [Batista et al. 2024]. A mudança de representação é justamente o que viabiliza analisar canais informais: listas de palavras-chave só reconhecem termos já cadastrados e não enxergam tecnologias sem vocabulário consolidado, que são as que mais interessam num estudo sobre competências emergentes. O mais próximo dessa linha é o BERTopic [Grootendorst 2022], que junta embeddings, redução de dimensionalidade e agrupamento por densidade a uma variante de TF-IDF por classe. Para o português, há modelos monolíngues como o BERTimbau [Souza et al. 2020].

Este artigo propõe um framework para extrair e analisar perfis de desenvolvedores em um grupo brasileiro do Telegram. Antes da vetorização, um filtro semântico em duas fases trata o ruído conversacional típico da mensageria. O número de agrupamentos vem de critérios de admissibilidade definidos antes da execução e arbitrados por um avaliador automático, e não de índices geométricos isolados, que premiam partições grosseiras. A rotulação e a auditoria ficam a cargo de modelos de famílias diferentes, seguindo o LLM-as-a-Judge [Zheng et al. 2023], de modo que nenhum avalie a própria saída — o que, como mostra a Seção 3.2, separa validação de autoconfirmação.

2. METODOLOGIA

2.1 Coleta e Filtragem de Dados

O conjunto bruto reúne 22.243 mensagens de um grupo público do Telegram voltado à divulgação de vagas e trabalhos autônomos em Front-End, Back-End e DevOps, entre maio de 2021 e janeiro de 2026. A exportação saiu do Telegram Desktop em formato JSON. A análise se restringe ao texto das mensagens: nomes de usuário, identificadores de remetente e metadados de autoria foram descartados ainda no parsing. O material é composto por anúncios publicados para divulgação em canal aberto, e não conversas privadas. Telefones, e-mails e URLs pessoais que restaram no texto viraram marcadores antes de qualquer processamento, e todos os resultados aparecem agregados por agrupamento ou por período.

A filtragem teve duas etapas. A Fase A remove mensagens de serviço, conteúdos vazios ou só com mídia, publicações com menos de 20 caracteres, mensagens formadas apenas por URLs, duplicatas de repostagem e textos sem predominância de caracteres latinos. Na Fase B, cada mensagem restante vira um vetor denso pelo modelo paraphrase-multilingual-mpnet-base-v2 e é comparada aos centróides de dois conjuntos de protótipos escritos à mão: 12 exemplos de formatos recorrentes de anúncio, do estruturado com benefícios ao recado curto de recrutador, e 11 exemplos de não-vaga, com casos de fronteira propositais, como dúvidas técnicas que citam as mesmas tecnologias e mensagens de quem procura recolocação. O score subtrai da similaridade com as vagas metade da similaridade com o ruído. O corte saiu da própria distribuição dos scores: na ausência de bimodalidade clara, adotou-se o percentil 25, o que levou a 0,188. Sobraram 7.264 registros, ou 32,7% do total, distribuídos em 47 meses entre setembro de 2021 e janeiro de 2026, dos quais 45 têm volume suficiente para a análise da Seção 3.3. A Tabela I detalha cada etapa.

As mensagens passaram por limpeza voltada a reduzir ruído sem perder o contexto técnico: normalização Unicode, remoção de URLs, anonimização e tratamento de emojis, com preservação dos caracteres de "C++", "C#" e "Node.js". A taxonomia de competências reúne 108 entradas em dez categorias técnicas — linguagens, frameworks, cloud, bancos de dados, DevOps, ERP, segurança, qualidade e testes, dados e BI, infraestrutura —, mais senioridade e regime de trabalho, cada uma com vários padrões para cobrir sinônimos e informalidades do mercado brasileiro. O casamento usa lookaround, e não fronteiras de palavra, por dois motivos: fronteiras falham em identificadores terminados em símbolo, o que impede reconhecer "C#" e "C++"; e a delimitação estrita é necessária em

Tabela I. Fluxo de filtragem, limpeza e seleção do corpus.

Etapa de Filtragem	**Registros Afetados**	**Justificativa**
Dataset Bruto	22.243	—
F1: `type` ≠ `message`	9.193	Remoção de ruído estrutural do Telegram
F2 e F3: Texto vazio ou < 20 caracteres	2.703	Mensagens sem conteúdo descritivo
F4 a F7: Links, mídias e duplicatas	662	Compartilhamento de links, imagens e repostagens
Fase A (Candidatos Semânticos)	**9.685**	—
Fase B: Semantic Score < 0.188	2.421	Remoção de ruído semântico (falsos positivos)
Dataset Final (Vagas Válidas)	**7.264**	**Corpus de análise (Taxa de retenção: 32.7%)**

português, onde termos técnicos aparecem dentro de palavras comuns, como "Scala" em "escalabilidade" ou "Express" em "expressão". Siglas ambíguas de duas letras ficaram de fora.

2.2 Representação Vetorial e Clusterização

Três modelos de representação foram avaliados quanto ao Silhouette médio no espaço de embedding: *paraphrase-multilingual-mpnet-base-v2* (0,1317), *intfloat/multilingual-e5-base* (0,1380) e o modelo monolíngue *neuralmind/bert-base-portuguese-cased*, BERTimbau (0,1940). O BERTimbau lidera o índice, mas foi descartado. Aplicado ao pipeline completo, ele gerou uma partição degenerada, organizada por idioma e por formato de redação em vez de perfil: dois agrupamentos ficaram com 78,6% do corpus e o avaliador reprovou a maior parte dos rótulos. Isso é coerente com a anisotropia de encoders treinados só com modelagem de linguagem mascarada, que comprime as similaridades cosseno e inflaciona medidas de compacidade sem ganho de separação semântica [Reimers and Gurevych 2019]. Adotou-se o *multilingual-e5-base*, dada a presença recorrente de anúncios em inglês no corpus.

A redução seguiu duas etapas. Primeiro PCA sobre os embeddings normalizados, com 30 componentes, que somam 46,5% da variância. A curva acumulada é plana: são precisos 37 componentes para 50% e 98 para 70%, e não há cotovelo a identificar. O valor de 30 corresponde ao piso mínimo adotado como salvaguarda, e o PCA serve aqui para atenuar ruído e acelerar o passo seguinte, não para preservar variância. Depois UMAP em duas configurações, ambas com n_neighbors = 15, métrica cosseno e semente 42: duas dimensões com min_dist = 0,1 para visualização, e dez dimensões com min_dist = 0,0 para o agrupamento.

A escolha do número de agrupamentos combina métricas internas e critérios definidos antes da execução. Silhouette e Davies-Bouldin medem separação geométrica e favorecem partições grosseiras, já que menos grupos produzem fronteiras mais nítidas. Aplicados sozinhos à varredura de K entre 5 e 24, com 50 inicializações por configuração, esses índices apontam partições em que um único agrupamento fica com mais da metade do corpus: ótimo pela geometria e sem utilidade descritiva, porque um perfil que abrange metade dos anúncios não é um perfil. Consideram-se admissíveis apenas as partições em que nenhum agrupamento passa de 25% do corpus e nenhum fica abaixo de 30 documentos, piso que a modelagem de tópicos por agrupamento exige. Entre os quatro melhores candidatos dentro desse conjunto, a decisão coube ao avaliador da Seção 2.3: o ciclo completo de tópicos, rotulação e auditoria rodou para cada um, e prevaleceu o de maior nota média (Tabela II).

Tabela II. Partições Admissíveis e Arbitragem

K	**Silhouette ↑**	**Davies-Bouldin ↓**	**Maior agrupamento**	**Coerência C_v**	**Escore médio do juiz**	**Aprovados**
18	0,4941	0,6840	10,4%	0,538	0,568	27,8%
19	0,4923	0,6911	9,1%	0,540	0,584	21,1%
20	0,4985	0,6986	9,0%	0,531	0,586	30,0%
23	0,5008	0,7197	8,9%	0,522	0,591	34,8%

Os quatro candidatos ficaram praticamente empatados, entre 0,568 e 0,591, com amplitude menor que o erro-padrão da média (0,035). O procedimento não mostra que K = 23 seja melhor que K = 18. Mostra que qualquer partição admissível serve sob o critério de interpretabilidade, e o máximo

funciona como regra de desempate. Vale notar que as partições admissíveis têm Silhouette menor que as inadmissíveis, e a queda acontece justamente onde os agrupamentos maiores se dividem. Era o esperado: separar um agrupamento heterogêneo produz fronteiras menos nítidas, ainda que os grupos resultantes digam mais. A opção foi por detalhe descritivo, e não por compacidade geométrica.

Na comparação entre métodos, o HDBSCAN, com hiperparâmetros escolhidos por DBCV máximo numa grade de vinte combinações, alcançou Silhouette de 0,5733 e Davies-Bouldin de 0,6288 sobre os pontos atribuídos, melhores que os 0,5001 e 0,7197 do K-Means com K = 23. A vantagem, porém, é condicional: ele marcou 19,7% dos anúncios como ruído, então suas métricas descrevem só os quatro quintos do corpus que aceitou particionar. O K-Means teve Calinski-Harabasz maior (14.033,0 contra 12.080,7) e cobre todos os registros, o que a análise da Seção 3.3 exige, já que descartar um em cada cinco anúncios distorceria as séries das competências menos frequentes.

2.3 Rotulação e Validação (*LLM-as-a-Judge*)

A caracterização de cada agrupamento junta extração estatística de tópicos e interpretação por modelo de linguagem. O LDA roda por agrupamento, com o número de tópicos entre 2 e 7 escolhido pela maior coerência C_v, métrica que acompanha melhor o julgamento humano sobre legibilidade. Nos 23 agrupamentos, a coerência média ficou em 0,522, entre 0,326 e 0,796. A rotulação coube ao *claude-haiku-4-5*, que recebeu por agrupamento os metadados do grupo, as competências mais frequentes, os termos do LDA, amostras representativas e uma rubrica de saída em JSON. O pipeline calcula o lift de cada competência em relação ao corpus completo e, quando nenhuma se destaca, instrui o modelo a nomear pela família de cargo ou pelo formato do anúncio, em vez de inventar especificidade técnica.

A auditoria seguiu o LLM-as-a-Judge [Zheng et al. 2023] em arquitetura cruzada: julgamento foi pelo *gpt-4o-mini*, de família e provedor distintos do rotulador. A separação responde a uma limitação conhecida do protocolo, já que um modelo que avalia o próprio rótulo mede consistência interna, não validade. O juiz desconhece a origem do rótulo e recebe apenas o rótulo, a descrição, a distribuição de competências recalculada sobre os documentos, amostras do agrupamento e os demais rótulos, estes para julgar se o nome distingue um agrupamento dos outros. Cada dimensão recebe uma de quatro âncoras (0,2, 0,5, 0,8 e 1,0), exemplificadas no prompt. A escala discreta reduz a variação entre execuções e contém a tendência dos modelos de serem generosos em escala aberta [Zheng et al. 2023], ao custo de menos granularidade.

A nota composta pondera três dimensões do rótulo: especificidade do nome (0,35), representatividade das competências (0,45) e qualidade da descrição (0,20). A coerência semântica fica fora da composição e é reportada à parte, por um motivo prático: ela avalia o agrupamento, não o rótulo. Um rótulo honesto dado a um agrupamento heterogêneo seria punido por um defeito que rótulo nenhum resolve, o que misturaria qualidade da partição com qualidade da rotulação. A nota composta é calculada em código, não pelo modelo, e rótulos com 0,70 ou mais são considerados aprovados. Prompts e saídas brutas de cada candidato de K estão no repositório do projeto[1].

2.4 Configuração Experimental e Uso de IA Generativa

Os experimentos rodaram no Google Colab, com semente global 42. O conjunto de dados não é redistribuído porque envolve mensagens de terceiros, mas o pipeline roda sobre corpus equivalente exportado do Telegram. Os modelos de linguagem entram apenas como instrumento, nas duas funções descritas na Seção 2.3, e não como autores: todas as decisões de projeto, da filtragem à interpretação dos resultados, cabem aos autores, que revisaram manualmente as saídas. Na redação, ferramentas de IA generativa foram usadas em revisão de escrita, sem gerar conteúdo científico. O corpus foi anonimizado antes do processamento e nenhum dado pessoal chegou a serviços de terceiros: as requisições

[1]Disponível em: https://doi.org/10.5281/zenodo.22101154

às APIs levam apenas competências agregadas e mensagens já anonimizadas. Por lidar só com dados públicos, agregados e não identificáveis, o estudo não configura pesquisa com seres humanos.

3. RESULTADOS E DISCUSSÃO

3.1 Descoberta de Perfis

A partição final produziu 23 perfis ocupacionais com distribuição equilibrada: o maior agrupamento reúne 644 anúncios (8,9% do corpus) e o menor, 59 (0,8%), com entropia normalizada de 0,969. A Figura 1 apresenta a topologia no espaço UMAP bidimensional e a Tabela III consolida os perfis com coerência e resultado da auditoria.

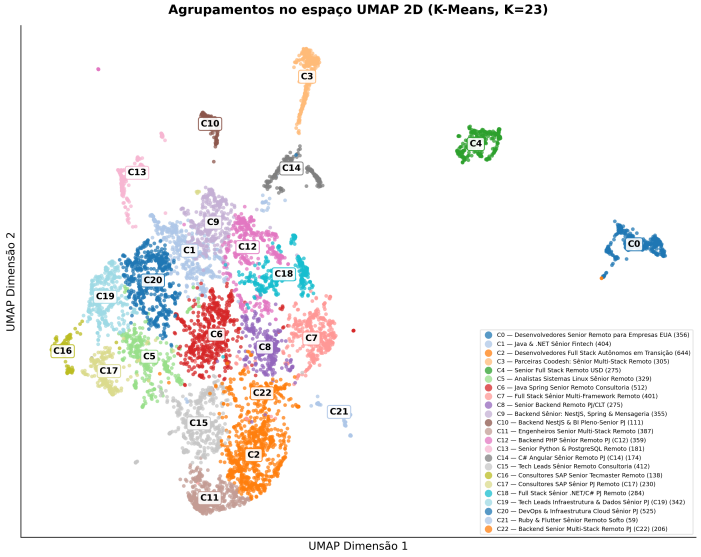

Fig. 1. Distribuição e Nomeação dos Clusters no Espaço UMAP 2D

Tabela III. Distribuição e Perfil Semântico dos Clusters, Métricas e Veredito

ID	Perfil	n	%	Stack Dominante	C_v	Composto	Veredito
C0	Desenvolvedores Sênior Remoto para Empresas EUA	356	4,9%	nextjs, swift, cpp, go, django	0,541	0,410	reprovado
C1	Java e .NET Sênior em Fintech	404	5,6%	springboot, kafka, sqlserver, oracle	0,353	0,605	reprovado
C2	Desenvolvedores Full Stack Autônomos em Transição	644	8,9%	—	0,326	0,620	reprovado
C3	Parceiras Coodesh: Sênior Multi-Stack Remoto	305	4,2%	c#, ruby, php, flutter, vue	0,541	0,605	reprovado
C4	Sênior Full Stack Remoto com Remuneração em USD	275	3,8%	typescript, python, react, postgresql	0,678	0,605	reprovado
C5	Analistas de Sistemas Linux Sênior Remoto	329	4,5%	linux	0,491	0,470	reprovado
C6	Java Spring Sênior Remoto em Consultoria	512	7,0%	java, springboot, oracle, kotlin	0,469	0,800	aprovado
C7	Full Stack Sênior Multi-Framework Remoto	401	5,5%	angular, react, c#, sqlserver	0,398	0,605	reprovado
C8	Senior Backend Remoto PJ/CLT	275	3,8%	react, vue	0,577	0,410	reprovado
C9	Backend Sênior: NestJS, Spring e Mensageria	355	4,9%	nestjs, springboot, kafka, rabbitmq	0,388	0,710	aprovado
C10	Backend NestJS e BI Pleno-Sênior PJ	111	1,5%	nestjs, powerbi, python, typescript	0,404	0,800	aprovado
C11	Engenheiros Sênior Multi-Stack Remoto	387	5,3%	—	0,433	0,260	reprovado
C12	Backend PHP Sênior Remoto PJ	359	4,9%	php, mysql, git, typescript, docker	0,506	0,710	aprovado
C13	Sênior Python e PostgreSQL Remoto	181	2,5%	python, postgresql	0,796	0,605	reprovado
C14	C# e Angular Sênior Remoto PJ	174	2,4%	c#, angular, kotlin, go, sap	0,635	0,605	reprovado
C15	Tech Leads Sênior Remoto Consultoria	412	5,7%	—	0,343	0,170	reprovado
C16	Consultores SAP Senior Tecmaster Remoto	138	1,9%	sap	0,765	0,605	reprovado
C17	Consultores SAP Sênior PJ Remoto	230	3,2%	sap	0,398	0,710	aprovado
C18	Full Stack Sênior .NET/C# PJ Remoto	284	3,9%	c#, mysql, sqlserver, mongodb	0,701	0,800	aprovado
C19	Tech Leads de Infraestrutura e Dados Sênior PJ	342	4,7%	linux, oracle, sqlserver, redes	0,653	0,710	aprovado
C20	DevOps e Infraestrutura Cloud Sênior PJ	525	7,2%	terraform, kubernetes, azure, jenkins	0,372	0,800	aprovado
C21	Ruby e Flutter Sênior Remoto	59	0,8%	ruby, flutter, react	0,683	0,605	reprovado
C22	Backend Sênior Multi-Stack Remoto PJ	206	2,8%	javascript, java, python, c#	0,563	0,365	reprovado
Média / Taxa de Aprovação					**0,522**	**0,591**	**8/23**

Três características emergem. A primeira é a uniformidade contratual: 22 dos 23 perfis são predominantemente remotos e 20, sêniores. O grupo analisado funciona sobretudo como canal de recrutamento remoto para gente experiente, e neste corpus senioridade e regime não separam um perfil de outro, razão pela qual não aparecem como colunas na Tabela III.

A segunda é a existência de perfis com contorno técnico claro, nas regiões mais isoladas do espaço vetorial: consultoria em SAP (C17), com lift máximo de 33,6; desenvolvimento remoto para empresas norte-americanas (C0, lift 16,1); infraestrutura, redes e dados (C19, lift 14,3); mensageria assíncrona com Kafka e RabbitMQ (C9, lift 9,0); e DevOps multi-cloud com Terraform e Kubernetes (C20). Java com Spring atravessa vários agrupamentos, e o segmento de sistemas legados e infraestrutura, com SAP, Oracle, SQL Server e redes, ocupa quatro deles, que taxonomias centradas em linguagens modernas dificilmente capturariam.

A terceira é a mais relevante do ponto de vista metodológico. Três agrupamentos não têm eixo tecnológico algum. C2, C11 e C15 somam 1.443 anúncios, ou 19,9% do corpus, e são os únicos em que nenhuma competência passa de lift 1,0, ou seja, nenhuma tecnologia está sobre-representada. Eles juntam autopromoção de candidatos, anúncios genéricos de liderança técnica sem stack declarada e ofertas multi-stack sem especialização. A proximidade vetorial vem de convenções de escrita, não de conteúdo técnico. Isso não é falha do agrupamento, e sim característica do corpus, isto é, cerca de um quinto dos anúncios de um canal informal não se organiza por perfil ocupacional. Protocolos de descoberta aplicados a fontes assim precisam separar agrupamentos formados por conteúdo daqueles formados por formato.

A modelagem de tópicos confirma essa leitura por outro caminho. A correlação entre tamanho do agrupamento e coerência C_v é de -0,638: o terço mais volumoso tem coerência média de 0,385, contra 0,606 do terço menor. Os agrupamentos grandes deste corpus não são nichos populosos, são resíduos, e absorvem o que não se encaixa nos nichos bem definidos, a modelagem de tópicos detecta isso independentemente do avaliador.

3.2 Auditoria dos Rótulos

A auditoria por *gpt-4o-mini* sobre rótulos de *claude-haiku-4-5* produziu escore composto médio de 0,591 (desvio-padrão 0,172), com 8 dos 23 rótulos aprovados, 9 na faixa intermediária e 6 abaixo do limiar crítico de 0,50. Por dimensão, as médias foram 0,748 em coerência semântica do agrupamento, 0,722 em qualidade da descrição, 0,565 em especificidade do nome e 0,552 em representatividade das competências.

A dispersão das notas diz mais que a média. Execuções preliminares do mesmo pipeline em autoavaliação, com um único modelo gerando e julgando, concentraram quase todas as notas em um só valor, o que indica falta de discriminação e não qualidade uniforme. A avaliação cruzada espalha as notas entre 0,17 e 0,80 e concentra as reprovações onde existe razão material: das 15 reprovações, 10 têm a especificidade do nome como pior dimensão e 5 a representatividade das competências, e nenhuma foi motivada pela descrição. Em outras palavras, o rotulador descreve bem e nomeia mal, ao produzir descrições fiéis ao conteúdo mas nomes que serviriam a qualquer outro agrupamento.

As piores notas caem exatamente nos agrupamentos sem eixo técnico: C15 (0,17), C11 (0,26) e C22 (0,365). Nos dois primeiros, o juiz puniu o rótulo por atribuir especificidade que a evidência não sustenta. O terceiro é mais curioso: o próprio rotulador registrou, em campo específico da saída, que quatro das competências que havia proposto não tinham suporte na distribuição do agrupamento. O juiz, sem acesso a esse campo, chegou à mesma conclusão por conta própria.

A correlação entre tamanho do agrupamento e nota do rótulo é de -0,003, praticamente nula. Isso muda o quadro em relação às configurações descartadas na Seção 2.2, nas quais os agrupamentos maiores eram sempre os piores avaliados. O critério de admissibilidade é o que produz essa independência, porque impede partições em que um agrupamento absorve o corpus e, com isso, remove a

maior fonte de heterogeneidade interna. O juízo automático não substitui avaliação humana especializada, e a concordância entre a nota do juiz e a opinião de especialistas não foi medida.

3.3 Análise Temporal

O volume mensal de publicações do grupo varia bastante, de modo que séries montadas sobre contagens absolutas medem a atividade do canal e não a demanda do mercado. Por isso cada competência aparece como participação nas vagas do mês, isto é, a razão entre o número de anúncios que a citam e o total de anúncios válidos daquele mês, considerando apenas os meses com pelo menos 30 anúncios: 45 dos 47 meses com registros, de maio de 2022 a janeiro de 2026. Cada série recebeu uma regressão linear e, como o teste envolveu 50 competências ao mesmo tempo, aplicou-se a correção de [Benjamini and Hochberg 1995]. Sem esse ajuste, cerca de um quarto das séries pareceria significativa a 5% por acaso.

Tabela IV. Participação nas vagas mensais. Inicial e final são as médias dos seis primeiros e dos seis últimos meses da série; Δ é a diferença entre elas; inclinação é regressão sobre os 45 meses; p ajustado sobre as 50 séries testadas; exibem-se as 12 de maior variação.

Competência	**Part. inicial (%)**	**Part. final (%)**	**Δ (p.p.)**	**Inclinação (p.p./ano)**	R^2	p **ajustado**	**Tendência**
C++	2,83%	0,12%	$-2,71$	$-0,72$	0,396	0,0002	Declinante
Vue	6,51%	3,75%	$-2,75$	$-1,30$	0,270	0,0063	Declinante
Linux	7,89%	3,78%	$-4,11$	$-1,68$	0,217	0,0211	Declinante
C#	19,42%	13,60%	$-5,82$	$-1,89$	0,160	0,0573	Estável
Kafka	3,25%	6,61%	$+3,36$	$+0,91$	0,158	0,0573	Estável
Next.js	1,89%	4,46%	$+2,56$	$+0,84$	0,130	0,0590	Estável
Python	13,32%	18,55%	$+5,22$	$+1,75$	0,094	0,1162	Estável
Kubernetes	6,52%	8,63%	$+2,11$	$+0,90$	0,095	0,1162	Estável
Azure	8,61%	12,81%	$+4,20$	$+0,86$	0,073	0,1720	Estável
AWS	17,25%	19,84%	$+2,59$	$+0,51$	0,037	0,3815	Estável
JavaScript	32,96%	28,92%	$-4,04$	$-1,90$	0,087	0,1287	Estável
Java	23,92%	22,28%	$-1,63$	$-0,75$	0,019	0,5387	Estável

O quadro que sai daí é de um mercado estável. Vinte e sete das 50 séries variam positivamente e apenas três quedas resistem à correção: C++, que quase some dos anúncios (2,83% para 0,12%), Vue (6,51% para 3,75%) e Linux (7,89% para 3,78%). Nenhuma alta atinge significância, embora Python (13,32% para 18,55%), Azure (8,61% para 12,81%), Kafka (3,25% para 6,61%) e Next.js (1,89% para 4,46%) apresentem os maiores ganhos, com p ajustado entre 0,057 e 0,172, o que sugere adoção inicial e não tendência consolidada. As tecnologias de maior volume seguem onde estavam: JavaScript vai de 32,96% para 28,92% e Java de 23,92% para 22,28%, ambas sem significância. Mesmo o C#, com a maior variação absoluta (19,42% para 13,60%), não resiste à correção.

Esse resultado contraria o que se obtém ao normalizar as séries pelo total de cada competência em vez do volume mensal de anúncios, procedimento que gera inclinação negativa para quase toda competência sempre que a atividade do canal cai e sugeriria, de forma equivocada, um declínio geral do mercado. Em análises longitudinais de canais informais, a escolha do denominador pesa tanto quanto a do modelo.

4. CONCLUSÃO

Este trabalho mostrou que é possível transformar mensagens informais em informação estruturada sobre o mercado de trabalho em tecnologia. O pipeline proposto, com filtragem semântica em duas fases, representação vetorial multilíngue, escolha de granularidade por critérios de admissibilidade e validação por juiz de arquitetura cruzada, produziu uma taxonomia de 23 perfis de desenvolvedores, dos quais oito receberam rótulos aprovados pelo avaliador independente. A análise temporal, feita sobre participação mensal e com correção para múltiplos testes, aponta um mercado estável no período, ao contrário do declínio geral que aparece quando se usam contagens absolutas.

Além disso, índices geométricos internos não bastam nem para escolher a representação nem para fixar o número de agrupamentos: o modelo de maior Silhouette gerou a pior taxonomia, e o critério composto sozinho aponta partições em que um agrupamento absorve metade do corpus. A independência entre rotulador e avaliador faz diferença, pois em autoavaliação o protocolo produz notas uniformes que não discriminam, enquanto em arquitetura cruzada ele espalha as notas e localiza as falhas em agrupamentos identificáveis. E cerca de um quinto do corpus forma agrupamentos organizados por formato de anúncio, não por perfil ocupacional, algo que a coerência temática confirma pela correlação entre volume e coerência.

Entretanto, há duas limitações. A avaliação é analítica: mede a coerência semântica dos perfis, não o efeito deles sobre contratações. E os dados vêm de uma única comunidade, o que restringe a validade externa. Entre os trabalhos futuros estão a comparação entre a recuperação por embeddings e a busca por palavra-chave em consultas típicas de recrutadores, a inclusão de um baseline sob o mesmo protocolo de avaliação e a construção de um subconjunto anotado por especialistas, que permitiria medir o alinhamento entre juízo automático e humano.

REFERÊNCIAS

Alibasic, A., Upadhyay, H., Simsekler, M. C. E., Kurfess, T., Woon, W. L., and Omar, M. A. Evaluation of the trends in jobs and skill-sets using data analytics: A case study. *Journal of Big Data* 9 (1): 32, 2022.

Batista, J. V. C. d. R., de Meireles Costa, G., and de Freitas Jorge, E. M. Mecanismo de busca semântica baseado em word embeddings em dados do currículo lattes, programas de pós-graduação e grupos de pesquisa. In *Escola Regional de Computação Bahia, Alagoas e Sergipe (ERBASE)*. SBC, pp. 109–118, 2024.

Benjamini, Y. and Hochberg, Y. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal statistical society: series B (Methodological)* 57 (1): 289–300, 1995.

Boselli, R., Cesarini, M., Mercorio, F., and Mezzanzanica, M. Using machine learning for labour market intelligence. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, pp. 330–342, 2017.

Colace, F., Santo, M. D., Lombardi, M., Mercorio, F., Mezzanzanica, M., and Pascale, F. Towards labour market intelligence through topic modelling. In *Proceedings of the 52nd Hawaii International Conference on System Sciences*. pp. 5256–5265, 2019.

Debao, D., Yinxia, M., and Min, Z. Analysis of big data job requirements based on k-means text clustering in china. *PloS one* 16 (8): e0255419, 2021.

Doğan, Y., Dalkiliç, F., Kut, A., Kara, K. C., and Takazoğlu, U. A novel stream mining approach as stream-cluster feature tree algorithm: A case study in turkish job postings. *Applied Sciences* 12 (15): 7893, 2022.

Grootendorst, M. Bertopic: Neural topic modeling with a class-based tf-idf procedure. *arXiv preprint arXiv:2203.05794*, 2022.

Ozcan, S., Sakar, C. O., and Suloglu, M. Human resources mining for examination of r&d progress and requirements. *IEEE Transactions on Engineering Management* 68 (5): 1372–1387, 2020.

Reimers, N. and Gurevych, I. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*. pp. 3982–3992, 2019.

Sibarani, E. M. and Scerri, S. Scodis: Job advert-derived time series for high-demand skillset discovery and prediction. In *International Conference on Database and Expert Systems Applications*. Springer, pp. 366–381, 2020.

Siswipraptini, P. C., Warnars, H. L. H. S., Ramadhan, A., and Budiharto, W. Information technology job profile using average-linkage hierarchical clustering analysis. *IEEE Access* vol. 11, pp. 94647–94663, 2023.

Souza, F., Nogueira, R., and Lotufo, R. Bertimbau: pretrained bert models for brazilian portuguese. In *Brazilian conference on intelligent systems*. Springer, pp. 403–417, 2020.

Sun, L., Cao, X., and Su, D. Statistical analysis of online recruitment information based on text mining technology. In *2022 International Conference on Intelligent Transportation, Big Data & Smart City (ICITBS)*. IEEE, pp. 431–436, 2022.

Wibowo, Y. P. Exploring job vacancy topics and trends in indonesia using latent dirichlet allocation (lda) and exploratory data analysis (eda). *Journal of Artificial Intelligence and Engineering Applications (JAIEA)* 5 (1): 1778–1785, 2025.

Zheng, L., Chiang, W.-L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E., et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems* vol. 36, pp. 46595–46623, 2023.