Web Scraping e Inteligência Artificial: Aplicações, Evolução e Perspectivas

Henrriky Bastos[1], *Henrique Pires*[1]

[1]Instituto Federal de São Paulo (IFSP) — Campus São Paulo, Brasil

h.jhonny@aluno.ifsp.edu.br, henrique.santiago@aluno.ifsp.edu.br

Abstract

Este artigo faz uma revisão sobre o uso de *web scraping* em conjunto com Inteligência Artificial (IA), destacando sua importância para a ciência de dados. O *web scraping* é apresentado dentro do fluxo de Extract–Transform–Load (ETL), com foco na etapa de extração de dados, e também em aplicações mais recentes, como as arquiteturas de *retrieval-augmented generation* (RAG), que usam bases vetoriais para ampliar o conhecimento de modelos de linguagem de grande porte (LLMs). São discutidos ainda aspectos legais e éticos, considerando leis como a Lei Geral de Proteção de Dados (LGPD), além de aplicações práticas em diferentes níveis organizacionais: operacional, tático e estratégico. O trabalho mostra também a evolução do tema, desde as primeiras formas de automação até seu estágio comercial atual. Conclui-se que o *web scraping*, quando aliado à IA, não serve apenas para coletar dados, mas se torna um recurso estratégico para inovação e competitividade, reforçando sua relevância para a área de ciência de dados.

Index Terms: web scraping; inteligência artificial; modelos de linguagem de grande porte (LLMs); geração aumentada por recuperação (RAG); engenharia de dados; ETL; ciência de dados; bases vetoriais; extração de dados; conformidade; LGPD; governança de dados; automação; ética em dados; inovação tecnológica

1. Introduction

Desde o "boom" da internet, o volume de dados disponível a cada ano cresce de forma acelerada, impulsionado por redes sociais, comércio eletrônico, artigos, bibliotecas de vídeos, imagens, entre outras plataformas digitais. Presume-se que, em 2025, o mundo atingirá um marco de 181 zettabytes de dados gerados [1], o que demonstra o desafio de coletar, processar e transformar essa quantidade enorme de informações em conhecimento útil para os setores público e privado. Dessa forma, torna-se cada vez mais necessária a utilização de técnicas que automatizem a extração dessas informações, capazes de lidar com dados em larga escala.

O *web scraping* é uma técnica que permite a coleta automática de dados da Web, realizando a conversão de páginas em informações estruturadas que podem ser analisadas em diferentes contextos, de acordo com o propósito final [2]. Quando utilizado em conjunto com a Inteligência Artificial (IA), especialmente em modelos de linguagem de grande porte (LLMs), permite não apenas a extração, mas também análises avançadas, contextualização e suporte às tomadas de decisão.

Este artigo tem como objetivo analisar os conceitos de *web scraping* e sua relação com a IA, destacando suas aplicações práticas, evolução histórica, estágio comercial e perspectivas futuras. Além disso, são discutidos os aspectos legais e éticos relacionados ao uso dessas técnicas, considerando regulamentações locais, como a Lei Geral de Proteção de Dados (LGPD).

Com esse propósito, o artigo está organizado da seguinte maneira: após esta introdução, apresenta-se uma revisão teórica sobre *web scraping*, Inteligência Artificial e sua integração em pipelines de Extract–Transform–Load (ETL), bem como em arquiteturas de *retrieval-augmented generation* (RAG). Em seguida, abordam-se aplicações práticas nos diferentes níveis organizacionais. Por fim, apresentam-se as considerações finais e as referências utilizadas.

2. Revisão Teórica

2.1. Conceitos-base: *Web Scraping* x *Web Crawling*

Web scraping é o processo que automatiza a extração de conjuntos de dados específicos de páginas na internet, utilizando ferramentas que organizam essas informações em formato estruturado, o que facilita sua transformação e utilização por meio de linguagens de programação. Por outro lado, o *web crawling* é o processo ou algoritmo que, de forma automática, navega entre diferentes páginas interligadas na internet por meio de uma cadeia de *hyperlinks*, tendo como principal objetivo coletar o conteúdo dessas páginas para realizar tarefas como indexação de sites por mecanismos de busca ou treinamento de modelos de IA. Esse método é amplamente utilizado por grandes empresas como Google, Bing e OpenAI. Na prática, muitos *pipelines* utilizam um *crawler* para mapear páginas e um *scraper* para extrair os dados de interesse [3].

2.2. *Web Scraping* no pipeline de ETL

No contexto de ETL (*Extract, Transform, Load*), o *web scraping* atua principalmente na fase de extração, ou seja, inicialmente são obtidos dados de páginas HTML, APIs públicas, arquivos estáticos (CSV, JSON) e documentos (PDFs). Posteriormente, esses dados passam por transformações — como limpeza, normalização e enriquecimento — e são carregados em *data lakes*, *data warehouses* ou bancos de dados SQL e NoSQL. A literatura de modelagem e integração de dados destaca o ETL como um dos pilares para análises em ciência de dados [4], o que reforça a importância do *web scraping* para essa área.

2.3. IA e LLMs: dos dados brutos ao contexto e treinamento

A combinação de *web scraping* com Inteligência Artificial (IA) amplia significativamente as possibilidades de uso de dados. Quando inseridos em um *pipeline* de ETL adequado, os dados extraídos podem ser organizados e utilizados em treinamentos supervisionados e não supervisionados de modelos de

IA. No caso específico de modelos de linguagem de larga escala (LLMs), o conhecimento armazenado é paramétrico, ou seja, limitado aos dados utilizados em sua fase de treinamento. Essa característica dificulta a obtenção de fatos e fontes recentes. Para superar tais restrições, surgiram arquiteturas que integram memória externa consultável, conhecidas como *retrieval-augmented generation* (RAG), nas quais o *web scraping* desempenha papel essencial no fornecimento das informações que alimentam esse processo [5].

2.4. RAG e bases vetoriais

O *Retrieval-Augmented Generation* (RAG) é uma técnica que combina memória paramétrica e não paramétrica, permitindo que modelos de linguagem de larga escala (LLMs) consultem uma base de conhecimento externa além do que foi aprendido durante o treinamento original. Essa abordagem foi formalizada por Lewis et al. [5] como uma forma de melhorar a precisão e a contextualização das respostas geradas por LLMs.

Dessa maneira, o RAG representa uma evolução dos modelos puramente paramétricos, ao incorporar mecanismos de recuperação de informação que possibilitam integrar conhecimento dinâmico e verificável ao processo de geração. Esse tipo de arquitetura tem sido amplamente adotado em aplicações de IA que demandam atualidade e transparência das fontes, como assistentes corporativos, sistemas de suporte e mecanismos de busca semântica [6, 7].

3. Aplicações e Usos

3.1. Aplicações gerais e práticas

Antes de entrar por áreas específicas, é válido trazer uma breve contextualização. Quando a raspagem de dados é combinada a modelos de IA, ela deixa de ser "coleta bruta" e passa a sustentar decisões do dia a dia — de acompanhar preços e notícias a apoiar diagnósticos e políticas públicas. Sempre que houver canais oficiais ou repositórios abertos (por exemplo, a coleção pública Common Crawl para textos da web, os catálogos acadêmicos Crossref e OpenAlex, ou o observatório de indicadores da Organização Mundial da Saúde), a coleta fica mais estável, reprodutível e auditável; quando esses canais não existem, a raspagem controlada ainda pode gerar valor, desde que venha acompanhada de limites de requisição, registro de versões e checagens de qualidade.

No varejo on-line, a coleta automatizada apoia o acompanhamento de preços, disponibilidade e tendências de consumo. Estudos econômicos que construíram índices a partir de milhões de preços coletados na web mostraram que esses índices andam de perto com os indicadores oficiais e ajudam a ler o mercado com mais rapidez. Revisões metodológicas em periódicos de estatística também discutem como transformar esses preços raspados em séries consistentes, comparando fórmulas e cuidados de qualidade. No mercado financeiro, o uso de raspagem e IA aparece em dois eixos. Primeiro, na análise textual de documentos corporativos (como demonstrações e relatórios), que deu origem a dicionários e práticas de leitura automática hoje consolidadas em finanças. Segundo, no avanço dos "dados alternativos" (notícias, avaliações de produtos, sinais da web) para previsão, gestão de risco e monitoramento — tema que já conta com revisões amplas na literatura. Em paralelo, pesquisas recentes mostram que modelos modernos (incluindo redes em grafos) vêm ampliando o desempenho na detecção de fraude.

Em ciência de dados e pesquisa, repositórios abertos tornaram viável treinar e avaliar modelos em larga escala. A coleção Common Crawl, por exemplo, disponibiliza periodicamente textos coletados da web e é amplamente usada em pré-treinamento; para literatura científica, trabalhos sobre o OpenAlex e sobre o Crossref descrevem como esses catálogos estruturam obras, autores, citações e vínculos, o que permite bibliometria, busca semântica e recuperação de contexto com rigor. Na saúde, há duas frentes claras. Para suporte clínico com recuperação de evidências, surgiram levantamentos e protótipos que integram recuperação e geração (RAG) com bases biomédicas, discutindo benefícios e limites de precisão factual. Para vigilância epidemiológica, um marco foi o painel interativo da Universidade Johns Hopkins, descrito em revista médica de alto impacto, que integrou múltiplas fontes públicas para acompanhar a COVID-19 em tempo real — um modelo de organização de coleta, validação e abertura de dados em crises sanitárias.

Em governo e políticas públicas, a literatura acadêmica e de organismos internacionais mostra como dados raspados ajudam a medir preços, acompanhar mercados de trabalho (ex.: vagas on-line) e produzir indicadores com maior frequência. Trabalhos em periódicos e working papers de confiança discutem ganhos e limitações: representatividade, vieses de cobertura e técnicas para transformar páginas em séries válidas para análise econômica. Na mídia e comunicação, bases que agregam notícias em múltiplos idiomas são usadas para mapear eventos, lugares e o "tom" de coberturas ao longo do tempo, com documentação acadêmica do seu processo de codificação e atualização. Para entender consumo e confiança em notícias, o Digital News Report 2025 (Universidade de Oxford) oferece séries comparáveis entre países; e, no combate à desinformação, há levantamentos de referência que revisam métodos de checagem automatizada, conjuntos de dados e desafios de interpretabilidade.

Em síntese, em todos os domínios, o valor do web scraping aliado à IA depende de três cuidados práticos: priorizar fontes e estudos reconhecidos (artigos, relatórios técnicos de pesquisa), padronizar qualidade ao longo do ETL (registros de coleta, amostragem, validações) e delimitar riscos e escopo de uso (privacidade, direitos, atualização). Esses elementos transformam a coleta em evidência útil para produtos analíticos, modelos e decisões.

3.2. Usos organizacionais por nível

A aplicação do *web scraping* combinado ao uso de IA pode ser compreendida de forma mais estruturada quando analisada sob a ótica dos níveis organizacionais — operacional, tático e estratégico. Cada um desses níveis demanda informações com diferentes graus de detalhamento e propósito analítico que, quando integrados corretamente, são essenciais para transformar dados em inteligência organizacional.

No nível operacional, o objetivo está na automação de tarefas repetitivas e na eficiência de processos. O *web scraping* atua como ferramenta de coleta contínua e sistemática de dados que alimentam rotinas diárias, reduzindo a necessidade de trabalho manual. Entre as aplicações mais comuns estão o monitoramento de preços e estoques em *e-commerces*, a coleta de avaliações e comentários de clientes em plataformas e a geração automática de relatórios básicos de desempenho. Quando associado a modelos de IA, esses dados podem ser classificados e resumidos em um intervalo de tempo menor, permitindo respostas rápidas e atualizadas com base nas solicitações.

No nível tático, a raspagem de dados tem finalidade mais analítica. Os dados extraídos são normalmente integrados a sis-

temas de *Business Intelligence* (BI) e transformados em *dashboards* que mostram tendências de mercado, comportamento de clientes e desempenho da empresa. O uso de técnicas de aprendizado de máquina sobre os dados coletados permite identificar padrões e anomalias, enriquecendo relatórios gerenciais com *insights* preditivos.

No nível estratégico, a combinação de *web scraping* e IA assume uma função de suporte à inovação e à formulação de políticas corporativas de longo prazo. As informações coletadas em larga escala podem alimentar modelos de previsão (*forecasting*), análises de competitividade e mecanismos de detecção de tendências emergentes. Empresas de tecnologia, finanças e varejo, por exemplo, utilizam *pipelines* de *scraping* e modelos de linguagem para antecipar mudanças no comportamento do consumidor, movimentações regulatórias ou novas oportunidades de mercado. Além disso, bases raspadas e tratadas adequadamente podem servir como insumo para projetos de inovação em produtos e serviços baseados em dados, fortalecendo a vantagem competitiva e a cultura *data-driven* das organizações. Nesse sentido, o *web scraping* deixa de ser uma simples técnica de coleta e passa a representar um ativo estratégico que sustenta decisões de alto impacto e orienta o posicionamento corporativo no mercado.

Em suma, os usos organizacionais do *web scraping* aliado à IA se estendem por todos os níveis da hierarquia empresarial — da automação operacional à inteligência estratégica —, reforçando a importância dessa tecnologia como pilar da transformação digital e da competitividade baseada em dados.

3.3. Aplicações práticas com modelos de linguagem de grande escala (LLMs)

Com o advento da IA, principalmente dos modelos de linguagem de grande escala (LLMs), observou-se uma oportunidade dentro do mercado para utilizar essa ferramenta no apoio ao atendimento externo e interno das companhias, que tradicionalmente era realizado por funcionários com base em conhecimentos internos. No entanto, como esses modelos possuem conhecimento limitado aos dados de treinamento, sua aplicação prática tornou-se desafiadora, uma vez que dados privados das companhias não estariam disponíveis para que o modelo formulasse respostas adequadas. Para superar esses desafios, surgiram técnicas complementares que permitem adaptar os modelos aos contextos corporativos.

3.3.1. Ajuste fino (Fine-tuning)

A primeira técnica utilizada para adaptar as respostas de um modelo de LLM, é o "ajuste fino" ou "fine-tuning", que consiste no processo de adaptação de um modelo treinado previamente para tarefas ou casos de uso específicos. Sua ideia principal é de aproveitar o conhecimento existente do modelo como ponto de partida para aprender novas tarefas, o que permite uma otimização de custo por não precisar treinar um modelo do zero.

O ajuste fino é considerado uma técnica que utiliza-se do aprendizado supervisionado, uma vez que para realizar o treinamento de um modelo de LLM para um domínio específico, é necessário fornecer um conjunto de dados rotulados. É nessa etapa que o *web scraping* pode atuar como uma ferramenta essencial para realizar a obtenção das informações, que eventualmente podem não estar disponíveis de forma oficial. Seu papel entra diretamente na capacidade de extrair dados rotulados a partir de páginas web que estejam alinhadas com o objetivo final do modelo que será treinado, transformando eles em dados rotulados que são utilizados dentro do processo de *fine-tuning*.

A título de exemplo, podemos citar uma uma página web de perguntas e respostas de uma empresa, que contém dados rotulados (as respostas) e podem ser facilmente transformados para realizar o aprendizado supervisionado por meio do *fine-tuning*. Abaixo, temos um exemplo de conjunto de dados disponível na plataforma Hugging Face [8], que pode ser utilizado para realizar o ajuste fino de modelos de linguagem.

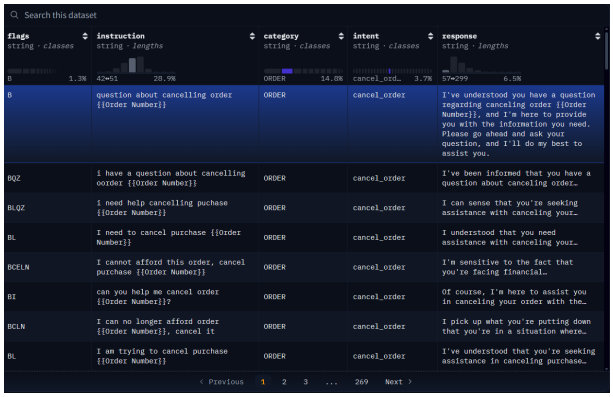

Figura 1: *Conjunto de dados do Hugging Face para fine-tuning de modelos de linguagem.*

Essa técnica costuma ser oferecida como serviço na nuvem, diretamente nos servidores das empresas que possuem um modelo de linguagem proprietário. A OpenAI, por exemplo, permite realizar o envio de arquivos CSV contendo a variável independente e a dependente (alvo). Após a execução do processo de *fine-tuning* com o conjunto de dados enviado, gera-se um modelo com identificador único que pode ser utilizado posteriormente, seja por meio de chamadas de API, para integrar com aplicações empresariais, ou pela própria interface da OpenAI. Por outro lado, existem LLM que são disponibilizados de forma gratuitas através de plataformas como o Hugging Face e que podem ser baixados em servidores dedicados, sendo possível aplicar o mesmo processo de fine-tuning que é utilizado dentro da plataforma de empresas que possuem modelos proprietários. Abaixo, temos um diagrama de como esse cenário poderia ser implementado:

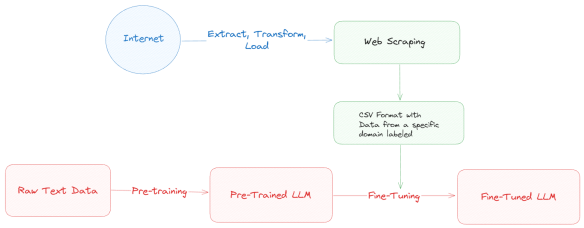

Figura 2: *Arquitetura típica para fine-tuning de modelos de linguagem com dados obtidos por Web Scraping.*

3.3.2. RAG (Retrieval-Augmented Generation)

O *Retrieval-Augmented Generation* (RAG) é um padrão de arquitetura no qual um modelo de linguagem (LLM) gera respostas condicionadas a uma base de conhecimento externa, armazenada em um mecanismo que permite a busca semântica de conteúdos. Esse acoplamento entre recuperação e geração reduz

o risco de alucinações, mantém a atualidade do conteúdo sem necessidade de treinamentos frequentes e possibilita a citação explícita das fontes utilizadas. Quando comparado ao *fine-tuning*, o RAG mostra-se mais adequado em contextos em que o conhecimento da organização é dinâmico, sujeito a alterações frequentes, ou quando há exigências de auditoria e restrições de acesso a dados sensíveis que não devem ser incorporados permanentemente ao modelo. O *fine-tuning*, por sua vez, mantém-se vantajoso em cenários nos quais se busca induzir estilo de escrita, formato de saída ou comportamentos específicos de tarefa, bem como generalizações a partir de exemplos rotulados. Na prática, ambas as abordagens são complementares: o RAG é mais apropriado para a gestão de conteúdo dinâmico e que possa conter citações, enquanto o *fine-tuning* é indicado para a adequação comportamental e de formato.

O *pipeline* típico do RAG inicia-se com a etapa de ingestão de conteúdo, na qual o *web scraping* desempenha papel fundamental na coleta automatizada de páginas autorizadas, como FAQs, políticas corporativas, manuais e documentações técnicas. O material coletado passa por processos de limpeza e normalização — por exemplo, conversão de HTML para texto puro ou Markdown —, sendo posteriormente enriquecido com metadados relevantes, como URL, título, data e versão. Essa etapa visa reduzir ruídos e garantir a consistência das informações que serão utilizadas.

Em seguida, realiza-se a segmentação textual, dividindo o conteúdo em passagens curtas com sobreposição controlada e alinhamento aos títulos das páginas. Essa técnica preserva a coesão textual e otimiza a etapa de recuperação semântica. Após a segmentação, cada trecho é convertido em uma representação vetorial (*embedding*), ou seja, uma forma numérica de representar o significado semântico do conteúdo. Essa transformação é realizada por modelos especializados, como o *text-embedding-3-large*, da OpenAI. Os vetores resultantes são armazenados em bancos de dados vetoriais — como Chroma, Pinecone, Weaviate e pgvector —, que permitem a recuperação semântica das informações com base na similaridade entre a consulta e os vetores armazenados. Entre as principais métricas utilizadas destacam-se a *Mean Reciprocal Rank* (MRR) e a similaridade de cosseno.

A última etapa do *pipeline* envolve a integração com o modelo de linguagem, na qual se implementa a lógica de consulta semântica e recuperação das informações a partir do banco de dados vetorial. Atualmente, muitos LLMs são treinados com suporte a chamadas de ferramentas (*tool calling*), mecanismo que permite ao modelo, durante a inferência, acionar de forma programática uma função externa previamente definida. Essa função pode executar uma busca semântica e retornar seus resultados, que são adicionados ao histórico de mensagens da sessão, servindo como contexto adicional para o modelo gerar uma resposta ao usuário final.

Nesse contexto, o *Retrieval-Augmented Generation* (RAG) faz uso direto desse conceito. Quando o usuário realiza uma consulta, o LLM invoca a ferramenta de busca semântica definida, passando como parâmetro a pergunta recebida. O banco de dados vetorial é então acionado e retorna um conjunto de trechos ranqueados conforme sua relevância. Com base nesse contexto recuperado, o LLM elabora uma resposta mais detalhada e precisa. Essa abordagem mostra-se especialmente útil em aplicações como atendimento ao cliente, treinamento automatizado em ambientes corporativos, suporte técnico e sistemas de perguntas e respostas sobre produtos ou processos internos [7].

Dessa forma, o *Retrieval-Augmented Generation* (RAG) representa um avanço significativo na integração entre recuperação de informação e geração de linguagem natural, permitindo que modelos de IA acessem e utilizem conhecimento atualizado, verificável e contextualizado, sem necessidade de treinamento constante. Abaixo, apresenta-se um diagrama ilustrativo do funcionamento dessa técnica.

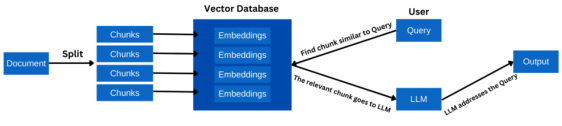

Figura 3: *Arquitetura típica do Retrieval-Augmented Generation (RAG) utilizando Web Scraping para coleta de dados.*

3.3.3. Extração de informações relevantes de páginas web

A extração de informações é uma das partes mais importantes do processo de *web scraping*, pois é nela que os dados coletados passam a ter valor prático. Normalmente, essa etapa é feita com regras fixas, como seletores CSS, expressões regulares e scripts que percorrem o HTML em busca de padrões conhecidos. O problema é que esse tipo de abordagem é bastante sensível a mudanças: qualquer alteração na estrutura da página pode quebrar o processo e exigir ajustes manuais. Isso faz com que projetos tradicionais de *scraping* sejam difíceis de manter quando há muitas fontes ou quando o conteúdo muda com frequência.

Com o avanço da Inteligência Artificial (IA), esse cenário começou a mudar. Modelos de linguagem de grande porte (*Large Language Models* — LLMs), como o GPT-4 e o Gemini 2.5, conseguem interpretar a estrutura e o contexto de uma página da web de maneira semelhante a um ser humano. Em vez de depender de seletores rígidos, é possível solicitar ao modelo que identifique, por exemplo, "todos os produtos e preços desta página" — e ele retorna apenas essas informações, mesmo que o HTML seja complexo ou mal formatado. Essa capacidade é fortalecida pelo uso dos chamados *structured outputs*, que permitem definir de antemão o formato da resposta (por exemplo, em JSON ou XML), garantindo que os resultados sejam estruturados e prontos para uso [9].

Além disso, o uso de técnicas de *function calling* e *prompt engineering* permite conectar o modelo diretamente a funções específicas dentro do *pipeline* de *scraping*. Isso faz com que a IA atue como uma camada intermediária de compreensão: ela interpreta o conteúdo e decide o que é relevante antes de enviar os dados para o banco. Em muitos casos, essa combinação reduz drasticamente o tempo de manutenção e melhora a qualidade do resultado, principalmente quando há muitas variações de layout entre as páginas.

Outro recurso importante é a aplicação de modelos voltados para extração de entidades nomeadas (*Named Entity Recognition* — NER). Eles são capazes de identificar automaticamente elementos como nomes, datas, valores e localidades dentro de um texto, o que ajuda a transformar dados brutos em informações realmente úteis. Esse tipo de técnica é bastante usado em cenários corporativos, por exemplo, para coletar informações de sites de concorrentes, portais de notícias, editais públicos ou bases científicas [10].

Em resumo, a IA trouxe uma nova camada de inteligência para o *web scraping*. Em vez de apenas "raspar" conteúdo, os sistemas passam a entender o que estão coletando. Isso torna todo o processo mais flexível, escalável e conectado a

aplicações mais complexas — como o *Retrieval-Augmented Generation* (RAG), análise de tendências e treinamento automatizado de agentes de IA baseados em dados atualizados.

4. Desafios de uso prático

4.1. Escalabilidade (custos computacionais)

4.2. Limitações técnicas

A qualidade dos dados extraídos por Web Scraping é um fator crítico que pode comprometer todo o pipeline subsequente. Dados capturados com erros de parsing, seletores desatualizados, formatações inconsistentes ou incompletas afetam diretamente a confiabilidade das inferências ou gerações realizadas por modelos de IA. Além disso, conteúdos dinâmicos ou carregados via JavaScript (AJAX, lazy loading) complicam a extração automática, exigindo ferramentas que simulem navegação ou monitorem as requisições de rede. Scrapers que não lidam com essas variações acabam extraindo dados incorretos ou vazios — um risco especialmente danoso em sistemas que dependem de recuperação vetorial (RAG) para gerar respostas acuradas.

Do ponto de vista da robustez e operacionalização, o sistema de scraping enfrenta desafios práticos como bloqueios anti-bot (CAPTCHAs, detecção de fingerprinting, bloqueio por IP), mudanças frequentes na estrutura dos sites, latência, falhas de rede e manutenção constante dos scrapers. Também há trade-offs entre frescor dos dados e custo: realizar scraping "ao vivo" pode tornar o sistema lento ou suscetível a falhas, enquanto fazer atualizações periódicas pode deixar o índice "desatualizado". Há ainda o risco de envenenamento de dados (inserção maliciosa) ou fragmentos irrelevantes, exigindo re-ranking ou filtros de validação. Em suma, construir um pipeline de Web Scraping integrado a IA requer não apenas técnicas sólidas de extração, mas uma arquitetura resiliente — capaz de detectar degradação, adaptar-se a variações e manter níveis aceitáveis de acurácia e disponibilidade ao longo do tempo.

4.3. Aspectos legais e éticos

O Web Scraping apresenta desafios legais significativos, envolvendo direitos autorais, termos contratuais de uso (termos de serviço) e regulamentações de proteção de dados pessoais. Mesmo que um site disponibilize conteúdo publicamente, a extração sistemática pode entrar em conflito com cláusulas que proíbem a raspagem ou com normas de privacidade, se houver coleta de informações identificáveis. Um exemplo emblemático é o caso hiQ Labs vs. LinkedIn: a hiQ raspava perfis públicos do LinkedIn para alimentar seus produtos de análise, mas o LinkedIn enviou notificação para barrar essa atividade, alegando violação de contratos e uso indevido de acesso. O Tribunal de Apelações da 9ª Região dos EUA inicialmente decidiu a favor da hiQ, entendendo que perfis públicos não são protegidos pela lei federal anti-hacking (CFAA) e que o bloqueio do LinkedIn poderia caracterizar prática anticompetitiva. Posteriormente, o caso voltou ao tribunal à luz de uma mudança interpretativa no âmbito da CFAA (decisão Van Buren) e acabou se encerrando em acordo, com hiQ reconhecendo que havia violado os termos de uso do LinkedIn e concordando em pagar indenização.

No plano ético, é fundamental agir com prudência e respeito a indivíduos e fontes. Recomenda-se anonimização ou agregação de dados, limitação de taxas de requisição (rate limiting), observância do protocolo robots.txt, transparência no propósito da coleta e salvaguarda dos direitos dos sites e dos usuários. Também é importante reconhecer que ética e legalidade se complementam — não basta estar legalmente "dentro da lei", é necessário que o uso dos dados seja responsável, respeitoso e consciente, minimizando riscos de danos e respeitando os limites implícitos de privacidade e propriedade intelectual.

5. Evolução do Tema e Estágio Comercial

Com a consolidação da Web e de seus padrões básicos (HTTP, HTML e URLs), surgiram os primeiros agentes automáticos capazes de percorrer hiperlinks e coletar conteúdo em escala. O crescimento acelerado de páginas e a necessidade de indexação impulsionaram a distinção entre descoberta (crawling) e extração (scraping), ainda que, nessa fase inicial, muitas soluções fossem rudimentares e voltadas a tarefas de arquivamento e busca. Essa base técnica foi viabilizada pelo próprio desenho aberto da Web e por sua padronização progressiva, coordenada por organismos como o W3C.

À medida que sites adotaram layouts mais ricos e conteúdo gerado dinamicamente, a extração migrou de scripts ad hoc para pipelines com etapas explícitas (coleta, parsing, normalização, validação). A literatura acadêmica começou a organizar o campo com classificações de técnicas (por exemplo, extração baseada em DOM, padrões e aprendizado), além de separar aplicações "corporativas" e "sociais". O foco passou a ser qualidade, robustez e reuso de componentes — um passo essencial para que a prática deixasse de ser artesanal e se tornasse parte do ecossistema de data engineering.

Com o avanço de aplicações ricas (AJAX, lazy loading), a extração precisou lidar com renderização de cliente, sessões e políticas anti-automação. Em paralelo, amadureceram diretrizes de acesso responsável (identificação de user-agent, rate limiting, back-off, respeito a políticas de acesso) e o próprio protocolo de exclusão de robôs, robots.txt, passou de convenção de fato a padrão formal: a RFC 9309 especifica linguagem, cache e tratamento de erros, reduzindo ambiguidades operacionais entre quem coleta e quem publica. Essa etapa consolidou um equilíbrio prático entre viabilidade técnica e governança do tráfego automatizado.

Com a maturidade dos pipelines, o mercado evoluiu de "raspar páginas" para entregar dados e conhecimento como serviços — índices pesquisáveis, catálogos temáticos e camadas de enriquecimento que abstraem a complexidade da coleta. Organizações passaram a decidir entre operar seus próprios pipelines ou consumir dados/infraestruturas de terceiros, avaliando custo de manutenção, cobertura, atualização (freshness) e requisitos legais. Na prática, a atenção deslocou-se do "como raspar" para SLA, governança e auditabilidade do que é entregue. (De forma geral, diretrizes de acesso e o padrão robots.txt seguem como referências para conformidade técnica).

A popularização de modelos de linguagem (LLMs) evidenciou um limite: o conhecimento interno ao modelo é paramétrico (aprendido no treino) e, portanto, envelhece e não carrega, por si, proveniência. A resposta arquitetural foi integrar recuperação externa ao processo de geração, a Retrieval-Augmented Generation (RAG), que combina memória paramétrica do LLM com memória não-paramétrica consultável em tempo de execução. Na prática, isso requer conteúdo bem coletado e estruturado (muitas vezes por pipelines de scraping), depois indexado/embarcado para recuperação semântica e citação de fontes. O resultado é um estágio abertamente comercial: provedores em nuvem documentam padrões e motores gerenciados de RAG, evidenciando que a coleta deixou de ser fim em si mesma para se tornar capacidade de atualização, checabilidade e contexto para aplicações de IA.

6. Conclusão

7. Page layout and style

The page layout should match with the following rules. A highly recommended way to meet these requirements is to use one of the templates provided and to check details against this example file. Do not modify the template layout! Do not reduce the line spacing!

If for some reason you cannot use any of the templates, please follow these rules as carefully as possible, or contact the organizers at <info@odyssey2026.org> for further instructions.

7.1. Basic layout features

- Proceedings will be printed in A4 format. The layout is designed so that the papers, when printed in US Letter format, will include all material but the margins will not be symmetric. PLEASE TRY TO MAKE YOUR SUBMISSION IN A4 FORMAT, if possible, although this is not an absolute requirement.
- Two columns are used except for the title part and possibly for large figures that may need a full page width.
- Left margin is 20 mm.
- Column width is 80 mm.
- Spacing between columns is 10 mm.
- Top margin is 25 mm (except for the first page which is 30 mm to the title top).
- Text height (without headers and footers) is maximum 235 mm.
- Page headers and footers must be left empty.
- No page numbers.
- Check indentations and spacing by comparing to the example PDF file.

7.1.1. Section headings

Section headings are centred in boldface with the first word capitalised and the rest of the heading in lower case. Sub-headings appear like major headings, except they start at the left margin in the column. Sub-sub-headings appear like sub-headings, except they are in italics and not boldface. See the examples in this file. No more than 3 levels of headings should be used.

7.2. Fonts

The font used for the main text is Times. The recommended font size is 9 points which is also the minimum allowed size. Other font types may be used if needed for special purposes. Remember, however, to embed all the fonts in your final PDF file!

LaTeX users: DO NOT USE THE Computer Modern FONT FOR TEXT (Times is specified in the style file). If possible, make the final document using POSTSCRIPT FONTS since, for example, equations with non-PS Computer Modern are very hard to read on screen.

7.3. Figures

Figures must be centred in the column or page (if the figure spans both columns). Figures which span 2 columns must be placed at the top or bottom of a page. Captions should follow each figure and have the format used in Fig. 4.

Figures should preferably be line drawings. If they contain gray levels or colors, they should be checked to print well on a high-quality non-color laser printer. If some figures contain bitmap images, please ensure that their resolution is high enough to preserve readability.

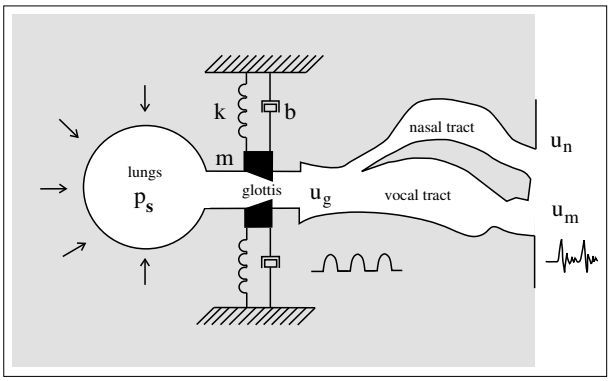

Figura 4: *Schematic diagram of speech production.*

7.4. Tables

An example of a table is shown in Table 1. Somewhat different styles are allowed according to the type and purpose of the table. The caption text may be above or below the table. Tables must be legible when printed in monochrome on A4 paper.

Tabela 1: *This is an example of a table*

Ratio	**Decibels**
$1/10$	-20
$1/1$	0
$2/1$	≈ 6
$3.16/1$	10
$10/1$	20

7.5. Equations

Equations should be placed on separate lines and numbered. Examples of equations are given below. Particularly,

$$x(t) = s(f_\omega(t)) \tag{1}$$

where $f_\omega(t)$ is a special warping function

$$f_\omega(t) = \frac{1}{2\pi j} \oint_C \frac{\nu^{-1k} d\nu}{(1-\beta\nu^{-1})(\nu^{-1}-\beta)} \tag{2}$$

A residue theorem states that

$$\oint_C F(z)dz = 2\pi j \sum_k Res[F(z), p_k] \tag{3}$$

Applying (3) to (1), it is straightforward to see that

$$1 + 1 = \pi \tag{4}$$

7.6. Page numbering

Final page numbers will be added later to the document electronically. *Please do not include any headers or footers!*

7.7. Style

Manuscripts must be written in English. Either US or UK spelling is acceptable (but do not mix them).

7.7.1. References

It is ISCA policy that papers submitted should refer to peer-reviewed publications. References to non-peer-reviewed publications (including public repositories such as arXiv, Preprints, and HAL, software, and personal communications) should only be made if there is no peer-reviewed publication available, should be kept to a minimum, and should appear as footnotes in the text (i.e., not listed in the References).

References should be in standard IEEE format, numbered in order of appearance, for example is cited before. For longer works such as books, provide a single entry for the complete work in the References, then cite specific pages or a chapter. Multiple references may be cited in a list.

7.7.2. International System of Units (SI)

Use SI units, correctly formatted with a non-breaking space between the quantity and the unit. In LaTeX this is best achieved using the `siunitx` package (which is already included by the provided LaTeX class). This will produce 25 ms, 44.1 kHz and so on.

8. Submissions

Information on how and when to submit your paper is provided on the conference website.

8.1. Manuscript

Authors are required to submit a single PDF file of each manuscript. The PDF file should comply with the following requirements: (a) no password protection; (b) all fonts must be embedded; and (c) text searchable (do ctrl-F and try to find a common word such as "the"). The conference organisers may contact authors of non-complying files to obtain a replacement. Papers for which an acceptable replacement is not provided in a timely manner will be withdrawn.

8.1.1. Embed all fonts

It is *very important* that the PDF file embeds all fonts! PDF files created using LaTeX, including on `https://overleaf.com`, will generally embed all fonts from the body text. However, it is possible that included figures (especially those in PDF or PS format) may use additional fonts that are not embedded, depending how they were created.

On Windows, the bullzip printer can convert any PDF to have embedded and subsetted fonts. On Linux & MacOS, converting to and from Postscript will embed all fonts:

```
pdf2ps file.pdf
ps2pdf -dPDFSETTINGS=/prepress file.ps file.pdf
```

9. Discussion

Authors must proofread their PDF file prior to submission, to ensure it is correct. Do not rely on proofreading the LaTeX source or Word document. **Please proofread the PDF file before it is submitted.**

10. Acknowledgements

The Odyssey 2026 organisers would like to thank ISCA and the organising committees of past Interspeech conferences for kindly providing the previous version of this template.

11. References

[1] F. Duarte, "Amount of data created daily (2025)," *Exploding Topics*, Apr. 2025, blog post. Accessed: 2025-09-22. [Online]. Available: https://explodingtopics.com/blog/data-generated-per-day

[2] C. Lotfi, S. Srinivasan, M. Ertz, and I. Latrous, "Web scraping techniques and applications: A literature review," in *Proc. SCRS Conference on Intelligent Systems*, India, Dec. 2022, accessed: 2025-10-11. [Online]. Available: https://www.publications.scrs.in/uploads/final_menuscript/863dc5628ae9215e611c22943d061742.pdf

[3] E. Ferrara, P. De Meo, G. Fiumara, and R. Baumgartner, "Web data extraction, applications and techniques: A survey," *Knowledge-Based Systems*, vol. 70, pp. 301–323, 2014, accessed: 2025-10-11. [Online]. Available: https://arxiv.org/pdf/1207.0246

[4] R. Kimball and M. Ross, *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling*, 3rd ed. Indianapolis, IN: Wiley, 2013, accessed: 2025-10-11. [Online]. Available: https://www.kimballgroup.com/data-warehouse-business-intelligence-resources/books/data-warehouse-dw-toolkit/

[5] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W. tau Yih, T. Rocktäschel, S. Riedel, and D. Kiela, "Retrieval-augmented generation for knowledge-intensive nlp tasks," in *Advances in Neural Information Processing Systems 33 (NeurIPS 2020)*, Vancouver, Canada, Dec. 2020, accessed: 2025-10-11. [Online]. Available: https://arxiv.org/abs/2005.11401

[6] Amazon Web Services, "What is rag? retrieval-augmented generation ai explained," AWS Official Documentation, Mar. 2025, accessed: 2025-10-19. [Online]. Available: https://aws.amazon.com/what-is/retrieval-augmented-generation/

[7] LangChain Documentation Team, "Rag: Retrieval-augmented generation in langchain," LangChain Official Documentation, Oct. 2025, accessed: 2025-10-22. [Online]. Available: https://docs.langchain.com/oss/python/langchain/rag

[8] Bitext and Hugging Face, "Bitext customer support llm chatbot training dataset," Hugging Face Datasets Repository, 2024, accessed: 2025-10-23. [Online]. Available: https://huggingface.co/datasets/bitext/Bitext-customer-support-llm-chatbot-training-dataset

[9] OpenAI, "Structured outputs documentation," OpenAI Official Documentation, 2024, accessed: 2025-10-18. [Online]. Available: https://platform.openai.com/docs/guides/structured-outputs

[10] IBM, "O que é reconhecimento de entidades nomeadas?" IBM Brasil — Think Blog, 2024, accessed: 2025-10-19. [Online]. Available: https://www.ibm.com/br-pt/topics/named-entity-recognition