



Relatório Final

arXplorer

Plataforma Web para Análise de Artigos do ArXiv

Text Mining

Discentes: João Dias nº 110305 /
David Franco nº 110733

Docente: Ana Catarina Martins

2024/2025

ÍNDICE

1. Introdução	1
2. Revisão de Literatura e Estado da Arte	3
2.1 Tecnologias e Conceitos Relevantes	3
2.2 Aplicações Semelhantes Existentes	4
2.3 Fundamentação Teórica e Técnicas Alternativas	4
3. Metodologia	6
3.1 Objetivos e Requisitos	6
3.1.1 Contextualização	6
3.1.2 Objetivo Geral	6
3.1.3 Objetivos Específicos	7
3.1.4 Requisitos Funcionais	8
3.1.4 Requisitos Não Funcionais	9
3.2 Arquitetura Geral da Aplicação	10
3.3 Backend da Aplicação	12
3.3.1 Obtenção e pré-processamento de dados	12
3.3.2 Análise de Palavras-Chave	13
3.3.3 Agrupamento Temático (Clustering)	15
3.3.4 Recomendação por Similaridade	16
3.4 Frontend da Aplicação	17
4. Resultados e Discussão	20
4.1 Demonstração funcional da UI	20
4.2 Deployment	25
4.3 Limitações e Melhorias Futuras	25
4.4 Comparação com Ferramentas Existentes	26
5. Conclusão	27
Referências	29

1. Introdução

Nas últimas décadas, o volume de publicações científicas tem crescido de forma acelerada, resultando numa sobrecarga de informação que torna cada vez mais difícil para investigadores e estudantes acompanharem os desenvolvimentos nas suas áreas de interesse (Bornmann & Mutz, 2015). Esta realidade levanta desafios não só ao nível da descoberta de novos artigos, mas também na análise das tendências emergentes, identificação de relações entre trabalhos e organização temática da literatura existente. Assim, a gestão eficaz deste conhecimento exige ferramentas que combinem automação, inteligência e interatividade (Dessì et al., 2020).

Neste contexto, surge o arXplorer, uma plataforma interativa construída com o intuito de apoiar a exploração e análise de literatura científica através de técnicas avançadas de text mining. O principal objetivo do projeto foi desenvolver uma aplicação capaz de: permitir a pesquisa avançada de artigos científicos segundo critérios definidos; extrair e visualizar tópicos e tendências predominantes nas publicações; identificar similaridades entre artigos com base em medidas de distância semântica, nomeadamente a similaridade do cosseno; e agrupar automaticamente os artigos em clusters temáticos com recurso ao algoritmo k-means, uma das abordagens mais populares para clustering de documentos (Jain, 2010).

A crescente complexidade e volume da produção científica têm evidenciado as limitações dos sistemas tradicionais de busca, que frequentemente se apoiam em correspondências literais de palavras-chave. Para superar essas restrições, têm sido propostas abordagens como técnicas de processamento de linguagem natural para a exploração da literatura científica (Schopf & Matthes, 2024). O arXplorer foi criado como uma ferramenta que integra análise semântica e visualização interativa, com o objetivo de facilitar a organização e a exploração de literatura científica por parte dos utilizadores e o seu repositório pode ver-se em:

<https://github.com/joaoprdias/arxplorer>.

Este relatório encontra-se organizado em seis secções principais. Após esta introdução, é apresentado o Estado da Arte, onde são revistos os conceitos teóricos e as tecnologias relevantes para o desenvolvimento da aplicação, incluindo técnicas de NLP e topic modeling. De seguida, na Metodologia, são descritos os dados utilizados, os processos de pré-processamento textual e os algoritmos implementados. A secção de Resultados apresenta os outputs obtidos com a aplicação, seguidos da Discussão, onde são analisadas as vantagens, limitações e implicações práticas da ferramenta. Por fim, na Conclusão, são sintetizados os principais contributos do projeto e sugeridas direções para trabalhos futuros.

2. Revisão de Literatura e Estado da Arte

2.1 Tecnologias e Conceitos Relevantes

A crescente quantidade de informação científica disponível tem impulsionado a necessidade de sistemas mais sofisticados para a sua organização, exploração e análise. Entre as tecnologias que mais têm contribuído para esta transformação destacam-se o Natural Language Processing (NLP), o Machine Learning (ML), os Large Language Models (LLMs) e os Knowledge Graphs.

Natural Language Processing é uma subárea da inteligência artificial dedicada à interação entre computadores e linguagem humana. No contexto científico, permite a extração automática de entidades, relações e significados a partir de textos complexos como artigos científicos. Este processamento pode ser realizado em diferentes níveis – desde a tokenização e identificação das classes gramaticais até à análise semântica e inferência – possibilitando a estruturação de informação não estruturada (*Young et al., 2018*).

Complementarmente, Machine Learning tem sido amplamente utilizado para construir modelos preditivos e de classificação sobre conjuntos de textos científicos, permitindo identificar padrões em larga escala, agrupar documentos semelhantes e sugerir recomendações personalizadas. Técnicas como clustering, classificação supervisionada, embeddings e análise de tópicos têm sido particularmente relevantes (*Kowsari et al., 2019*).

Nos últimos anos, os LLMs, como o BERT (Bidirectional Encoder Representations from Transformers), vieram revolucionar a forma como os sistemas computacionais comprehendem texto. Ao serem pré-treinados em grandes volumes de dados e adaptados a tarefas específicas como a análise de tópicos, extração de resumos ou identificação de entidades nomeadas, estes modelos permitem uma compreensão muito mais profunda do conteúdo textual (*Devlin et al., 2019*). Recentemente, estes modelos têm sido aplicados ao domínio científico, com variantes como SciBERT (*Beltagy et al., 2019*) ou TopicGPT (*Pham et al., 2024*), otimizadas para vocabulário técnico e relações interdisciplinares.

Outro conceito fundamental é o de Knowledge Graphs, que representa o conhecimento de forma estruturada através de nós (entidades) e arestas (relações entre entidades). Esta estrutura é particularmente eficaz para organizar informação científica, pois permite interligar conceitos, autores, tópicos e publicações, facilitando a navegação semântica e a descoberta de conhecimento implícito (*Hogan et al., 2021*). Os Knowledge Graphs podem ser construídos manualmente ou por meio de técnicas automáticas de extração de conhecimento a partir de textos, utilizando NLP e redes semânticas.

Além disso, ferramentas modernas têm vindo a combinar estes elementos. Por exemplo, sistemas como o Semantic Scholar usam BERT e Citation Graphs para explorar a literatura de forma semântica e contextual (Lo et al., 2020). A convergência de alguns destes métodos também está na base do arXplorer, permitindo superar as limitações das pesquisas por palavra-chave e promover uma exploração mais inteligente e significativa do conhecimento científico.

2.2 Aplicações Semelhantes Existentes

No setor do marketing digital e da análise de redes sociais, plataformas de social listening como o Brandwatch e o Talkwalker destacam-se pelo uso extensivo de NLP para extrair insights sobre sentimentos, tendências e padrões de comportamento a partir de texto não estruturado em tempo real. Estas ferramentas aplicam técnicas como análise de sentimentos, extração de tópicos e named entity recognition para monitorizar percepções públicas e dar suporte a decisões estratégicas em empresas (Batinca & Treleaven, 2015).

No domínio académico, soluções como o Semantic Scholar combinam NLP com redes neurais profundas para oferecer uma experiência de pesquisa semântica mais rica. Esta plataforma utiliza modelos como o SciBERT, um language model treinado especificamente em textos científicos, para extrair automaticamente keywords, sumarizar conteúdos e sugerir artigos relacionados com base em similaridade semântica (Beltagy, Lo, & Cohan, 2019).

Adicionalmente, aplicações mais recentes como o TopicGPT (Pham et al., 2024) ilustram o uso de prompt-based topic modeling com modelos de linguagem de grande escala (LLMs) para facilitar a exploração temática de textos académicos. Estes sistemas não apenas identificam tópicos relevantes, mas também permitem interação flexível através de prompts interpretáveis, o que os torna especialmente úteis em cenários exploratórios e pedagógicos.

Estas ferramentas demonstram como a integração de NLP com visualização, modelação semântica e user-centered design pode transformar o modo como os utilizadores interagem com informação textual complexa — uma visão que também guia o desenvolvimento do arXplorer.

2.3 Fundamentação Teórica e Técnicas Alternativas

A análise semântica da literatura científica tem evoluído significativamente com o avanço de técnicas de topic modeling, nomeadamente no que diz respeito à superação das limitações do modelo tradicional Latent Dirichlet Allocation (LDA). Embora o LDA tenha sido amplamente utilizado pela sua capacidade de identificar temas latentes em grandes coleções de texto, ele assume independência entre

palavras e carece de mecanismos eficazes para capturar relações semânticas mais complexas.

Nesse contexto, têm surgido modelos alternativos que procuram melhorar a coerência e a expressividade dos tópicos extraídos. Por exemplo, o GINopic recorre a técnicas de graph isomorphism networks (GIN) para representar relações entre palavras como grafos, permitindo capturar estruturas semânticas mais ricas e coerentes (*Adhya & Sanyal, 2024*).

Outra abordagem inovadora é o PromptTopic, que explora o potencial dos LLMs através de prompt engineering para realizar topic modeling com base em frases individuais. Esta estratégia tem demonstrado maior sensibilidade semântica em textos curtos e diversificados, superando limitações de esparsidade e ambiguidade frequentes em modelos clássicos (*Wang et al., 2023*).

O Neural Dynamic Focused Topic Model, por sua vez, introduz um modelo neural dinâmico que permite modelar a evolução temporal de tópicos em documentos sequenciais. Este tipo de abordagem é especialmente relevante para estudos longitudinais ou para o acompanhamento da mudança de temas ao longo do tempo (*Cvejoski et al., 2023*).

Além disso, a integração de modelos de linguagem pré-treinados como o BERT em arquiteturas de topic modeling, como no caso do BERTopic, tem permitido abordar de forma mais eficaz desafios associados a textos curtos, como publicações em redes sociais ou abstracts científicos. Esta combinação melhora a representatividade semântica e a agrupação de tópicos ao introduzir conhecimento contextual previamente aprendido (*Grootendorst, 2022*).

Estas abordagens refletem uma mudança de paradigma no campo do topic modeling. O arXplorer inspira-se nestes avanços para combinar técnicas modernas de análise de tópicos com visualizações interativas, promovendo uma exploração semântica mais intuitiva e informada da literatura científica.

3. Metodologia

3.1 Objetivos e Requisitos

3.1.1 Contextualização

Com a crescente quantidade de artigos científicos disponíveis em repositórios abertos, torna-se cada vez mais difícil para os utilizadores identificarem, organizarem e compreenderem o conhecimento disponível em grandes volumes de dados. A plataforma arXiv, criada em 1991, disponibiliza pré-publicações científicas nas áreas da física, matemática, ciência da computação, entre outras. Contudo, a sua interface base limita-se a funcionalidades de pesquisa textual e filtragem por campos simples, sem oferecer ferramentas exploratórias mais avançadas, que permitam fazer análise textual, agrupamento temático ou até recomendação de artigos.

Desta forma, existe uma lacuna na disponibilização de ferramentas acessíveis e interativas que permitam a análise exploratória de literatura científica de forma rápida e com reduzida carga cognitiva. Esta necessidade motivou o desenvolvimento da aplicação arXplorer, que procura aliar a simplicidade de utilização à capacidade analítica, recorrendo a técnicas de text mining.

3.1.2 Objetivo Geral

O objetivo geral da aplicação consiste no desenvolvimento de uma web app interativa que permita aos utilizadores explorar conteúdos científicos da plataforma arXiv através de técnicas de text mining, focando-se essencialmente na pesquisa, análise, agrupamento e recomendação de artigos. A aplicação pretende oferecer uma experiência integrada, permitindo não só consultar artigos relevantes, mas também compreender as principais temáticas abordadas e as relações existentes entre diferentes publicações.

Neste sentido, a aplicação tem como objetivo permitir a pesquisa de artigos com base em palavras-chave, a identificação automática dos termos mais frequentes nos resumos das publicações, o agrupamento dos artigos em tópicos temáticos recorrendo a algoritmos de clustering, e a recomendação de artigos semelhantes com base em proximidade semântica. Estas funcionalidades são complementadas com visualizações interativas que facilitam a exploração dos resultados por parte do utilizador.

A aplicação foi desenvolvida de forma a ser multiplataforma, garantindo compatibilidade com dispositivos móveis e desktop, e está disponível online através de serviços que asseguram toda a infraestrutura necessária para o deployment. Esta abordagem assegura o acesso aberto e imediato ao sistema, sem necessidade de instalação local.

Este objetivo surge da necessidade crescente de ferramentas acessíveis que apoiam a análise exploratória de grandes volumes de literatura científica. Ao combinar técnicas de text mining com uma interface web moderna e responsiva, pretende-se oferecer um contributo prático para a comunidade académica e científica na identificação e organização de conhecimento em contextos de investigação.

3.1.3 Objetivos Específicos

Para alcançar o objetivo geral definido na secção anterior, foram estabelecidos objetivos específicos, organizados de acordo com as diferentes componentes da aplicação, tais como:

- **Desenvolver uma arquitetura modular e escalável**

Estruturar a aplicação segundo o modelo cliente-servidor, separando de forma clara o frontend e o backend. Esta arquitetura permite isolar responsabilidades, garantir a escalabilidade da solução e facilitar futuras extensões funcionais.

- **Integrar a API pública do arXiv para recolha de dados científicos**

Implementar um mecanismo de comunicação com a API do arXiv que permita obter artigos científicos em tempo real, com base em parâmetros de pesquisa definidos pelo utilizador, tais como palavras-chave, ordenação por relevância ou data, e categorias científicas.

- **Aplicar técnicas de text mining ao conteúdo dos artigos**

Processar os resumos dos artigos obtidos, recorrendo a técnicas como normalização textual, remoção de stopwords, lematização e vetorização. Este processamento deve preparar os dados para a análise estatística e semântica, assegurando a consistência e a comparabilidade entre os textos.

- **Identificar automaticamente palavras-chave relevantes**

Extrair as unidades linguísticas mais representativas dos títulos e resumos dos artigos com base em contagem absoluta, considerando unigramas (termos isolados), bigramas (pares de palavras) e trigramas (sequências de três palavras), permitindo identificar não apenas palavras individuais, mas também expressões compostas com significado contextual. Os n-gramas mais frequentes fornecem ao utilizador uma visão condensada e estruturada dos temas predominantes no conjunto de artigos obtido.

- **Agrupar os artigos em tópicos com base na sua similaridade textual**

Implementar algoritmos de agrupamento para identificar automaticamente clusters temáticos. Os artigos devem ser representados como vetores num espaço semântico, e o número de clusters pode ser definido estaticamente.

- **Recomendar artigos semelhantes com base em proximidade semântica**

Dado um artigo selecionado, calcular a sua semelhança com os restantes artigos através de métricas como a similaridade do cosseno. A aplicação deverá apresentar ao utilizador os artigos mais semelhantes, ordenados por grau de similaridade, promovendo a descoberta de conteúdos relacionados.

- **Desenvolver uma interface web responsiva e interativa**

Construir um frontend intuitivo em React.js, garantindo a acessibilidade da aplicação em diferentes dispositivos. A interface deve permitir ao utilizador realizar pesquisas, visualizar os resultados de forma clara e explorar os agrupamentos temáticos através de uma visualização interactiva com D3.js.

3.1.4 Requisitos Funcionais

Os requisitos funcionais (RF) definem o conjunto de funcionalidades que a aplicação deve disponibilizar ao utilizador para cumprir os objetivos definidos nas secções anteriores. Estes requisitos foram identificados com base nas necessidades da análise exploratória de artigos científicos e refletem diretamente as funcionalidades implementadas na aplicação.

RF1 – Permitir a pesquisa de artigos na API pública do arXiv com base em palavras-chave e filtros;

RF2 – Permitir a personalização da ordenação dos resultados (por relevância ou por data);

RF3 – Apresentar os resultados da pesquisa em formato de lista com título, autores, data e resumo;

RF4 – Permitir a paginação dos resultados apresentados ao utilizador;

RF5 – Processar os títulos e resumos dos artigos recolhidos para análise textual posterior;

RF6 – Extrair automaticamente os n-gramas mais frequentes (unigramas, bigramas e trigramas) com base em contagem absoluta e apresentar ao utilizador os n-gramas mais frequentes sob a forma de lista ordenada por frequência;

RF7 – Representar cada artigo através de vetores numéricos para análise semântica;

RF8 – Agrupar os artigos automaticamente em clusters temáticos com o algoritmo K-Means;

RF9 – Permitir a visualização interativa dos clusters com zoom, drag-click e destaque de artigos;

RF10 – Identificar e apresentar os artigos pertencentes a cada cluster com metainformação relevante;

RF11 – Calcular a similaridade entre artigos com base na similaridade do cosseno;

RF12 – Apresentar ao utilizador uma lista de artigos semanticamente semelhantes a um artigo selecionado;

RF13 – Permitir a navegação entre páginas principais da aplicação: Pesquisa, Análise, Clustering e Recomendação;

RF14 – Permitir ao utilizador regressar à página inicial a partir de qualquer página da aplicação;

RF15 – Permitir ao utilizador selecionar o número de resultados a obter da API do arXiv;

RF16 – Apresentar feedback visual durante carregamento de dados;

RF17 – Apresentar mensagem de erro caso a pesquisa não devolva resultados;

RF18 – Permitir clicar num artigo para abrir a respetiva página do arXiv num novo separador;

Cada um destes requisitos foi implementado e testado no decorrer do desenvolvimento da aplicação, assegurando que o sistema corresponde às necessidades inicialmente identificadas e oferece uma experiência funcional e útil para os utilizadores.

3.1.4 Requisitos Não Funcionais

Os requisitos não funcionais (RNF) definem as características de qualidade da aplicação, ou seja, propriedades que não estão diretamente relacionadas com funcionalidades visíveis, mas que influenciam o desempenho, a experiência de utilização, a organização interna e a manutenção do sistema. No contexto do arXplorer, foram considerados os seguintes requisitos não funcionais:

RNF1 – Assegurar que a aplicação está disponível publicamente, sem necessidade de instalação local;

RNF2 – Publicar o frontend na plataforma Vercel e o backend na plataforma Render, com integração contínua via GitHub;

RNF3 – Manter a separação clara entre frontend e backend, comunicando exclusivamente via API RESTful;

RNF4 – Centralizar todas as chamadas à API do backend num único módulo, promovendo uma organização limpa do código;

RNF5 – Garantir que o estado da aplicação é preservado entre páginas durante a navegação do utilizador;

RNF6 – Utilizar bibliotecas de código aberto e amplamente suportadas, como React, FastAPI, scikit-learn e D3.js;

RNF7 – Adaptar a interface a diferentes tamanhos de ecrã, garantindo uma experiência responsiva em dispositivos móveis e desktop;

RNF8 – Organizar o código de forma modular e extensível, permitindo fácil manutenção e futura evolução da aplicação;

Estes requisitos foram tidos em consideração ao longo de todo o processo de desenvolvimento, garantindo que a aplicação é tecnicamente robusta, acessível e está preparada para evoluir de forma sustentável.

3.2 Arquitetura Geral da Aplicação

A aplicação arXplorer foi desenvolvida com base numa arquitetura modular do tipo cliente-servidor, garantindo uma separação clara entre a camada de apresentação (frontend) e a camada de lógica e dados (backend). Esta decisão permitiu isolar responsabilidades, promover a escalabilidade da solução e facilitar a sua manutenção e evolução. A Figura I ilustra de forma esquemática esta arquitetura, destacando os principais módulos responsáveis pelas operações da aplicação e os fluxos de comunicação entre componentes.

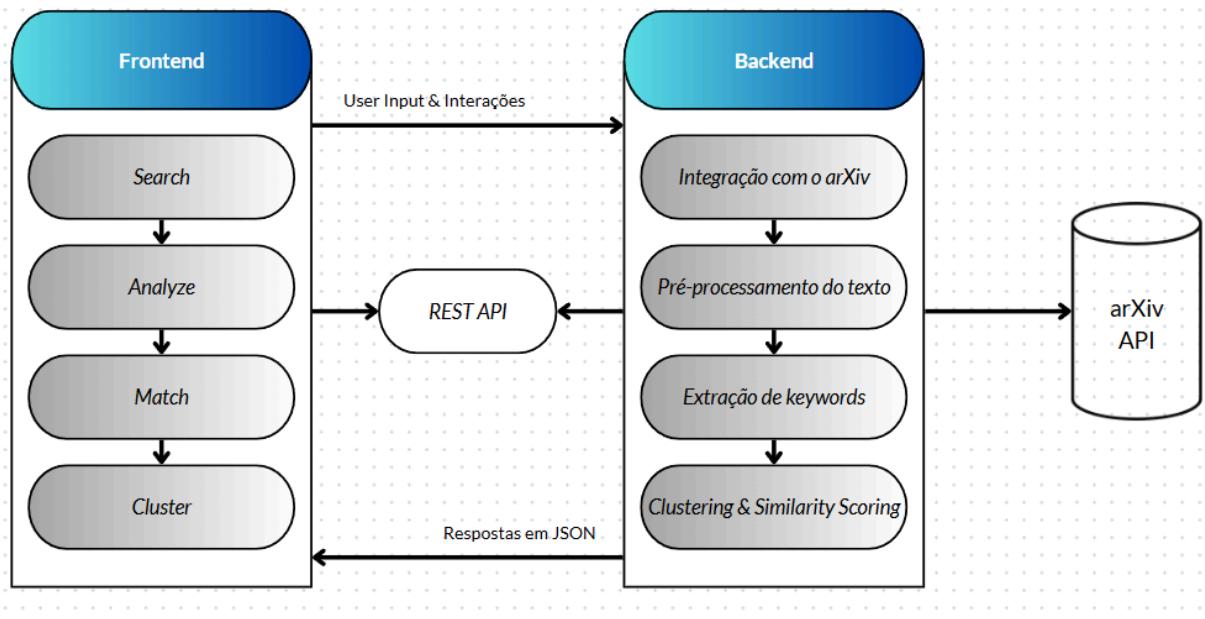


Figura I - Arquitetura Geral do arXplorer

O frontend, desenvolvido em React.js, é responsável pela interação com o utilizador, recolha de inputs (como palavras-chave de pesquisa), navegação entre funcionalidades, apresentação dos resultados e visualizações interativas. Toda a comunicação com o backend é feita através de chamadas a uma API RESTful, encapsuladas num único módulo (services/api.js) que centraliza as requisições e melhora a organização do código.

O backend foi desenvolvido em FastAPI e expõe um conjunto de endpoints RESTful que suportam todas as operações da aplicação. Estes endpoints incluem a recolha de artigos científicos a partir da API pública do arXiv, o pré-processamento dos textos recolhidos, a extração de n-gramas com base em contagem absoluta, a vetorização dos resumos, a aplicação do algoritmo de clustering K-Means, o cálculo da similaridade entre artigos com base na distância cosseno, e a devolução de respostas estruturadas ao frontend no formato JSON. Os dados são tratados em memória, uma vez que não existe necessidade de armazenamento persistente.

Para garantir acessibilidade e disponibilidade contínua, a aplicação foi publicada online com recurso a plataformas de deployment automático. O frontend foi alojado na Vercel, que permite integração direta com o GitHub e execução contínua de builds. O backend foi publicado na plataforma Render, que oferece suporte a aplicações FastAPI e gestão de instâncias de forma simplificada. Esta estratégia garante que a aplicação pode ser acedida por qualquer utilizador, sem

necessidade de instalação local ou configuração adicional. A aplicação encontra-se disponível publicamente através do endereço: <https://arxplorer.vercel.app>.

3.3 Backend da Aplicação

3.3.1 Obtenção e pré-processamento de dados

O ponto de partida de grande parte das componentes do arXplorer consiste na recolha dinâmica de artigos científicos através da API pública do arXiv. Esta API expõe metadados de artigos submetidos no repositório, permitindo que aplicações externas acedam a milhares de documentos em tempo real.

As respostas da API seguem o formato Atom, que tem uma estrutura hierárquica em XML que encapsula cada item de conteúdo (no caso, cada artigo) num elemento `<entry>`, contendo subelementos como `<title>`, `<summary>`, `<author>`, `<published>`, `<updated>`, `<id>` (URL do artigo), entre outros. A API do arXiv estende este padrão com elementos próprios, como `<arxiv:comment>`, `<arxiv:primary_category>` e `<arxiv:journal_ref>`, pertencentes a um namespace específico. O parsing destes documentos XML requer o reconhecimento dos namespaces e o mapeamento explícito de cada elemento relevante.

Na aplicação, cada chamada à API é construída a partir de uma expressão de pesquisa definida pelo utilizador, podendo incluir operadores booleanos e critérios de ordenação explícitos (por relevância ou data de submissão/atualização). A API impõe um limite de 1000 resultados por cada request, pelo que a aplicação gera automaticamente a paginação dos pedidos através dos parâmetros `start` e `max_results`, agregando os resultados num único conjunto de dados estruturado. O sistema extraí, de forma automática, campos como o título, o resumo, os autores, a data de publicação e a categoria principal de cada artigo, convertendo-os para uma estrutura de DataFrame compatível com as etapas posteriores de análise.

Após a recolha, segue-se o pré-processamento textual, que se trata de uma fase essencial para transformar o texto raw em representações linguísticas mais homogéneas e apropriadas para análise computacional. Esta etapa incide sobre os títulos e resumos dos artigos, que contêm o conteúdo semântico mais relevante para a extração de tópicos e deteção de padrões.

O pré-processamento inicia-se com a normalização lexical, onde o texto é convertido para minúsculas, e todos os sinais de pontuação, números e símbolos não alfabéticos são removidos por expressões regulares. Esta uniformização evita duplicações semânticas causadas por variações superficiais de capitalização ou pontuação.

Segue-se a tokenização, processo que divide o texto em unidades linguísticas discretas — geralmente palavras — denominadas tokens. A tokenização é um passo fundamental no processamento de linguagem natural, já que fornece a granularidade mínima sobre a qual são construídas representações vetoriais, medidas de frequência ou algoritmos de análise semântica.

Posteriormente, realiza-se a remoção de stopwords, isto é, palavras muito frequentes numa língua (como “the”, “and”, “of”, entre outras) que, pela sua ubiquidade, tendem a diluir o sinal informativo do texto. A eliminação destas palavras permite focar a análise nos termos que melhor permitem distinguir dois artigos científicos diferentes.

Por fim, aplica-se a lematização, uma técnica de normalização morfológica que converte cada palavra para o seu “lema”, ou forma base canónica. Diferentemente do stemming — que corta sufixos de forma heurística e pode gerar palavras truncadas ou inexistentes —, a lematização preserva a integridade lexical e semântica, agrupando variações morfológicas que partilham o mesmo significado. Este processo é especialmente importante para o contexto de investigação, onde a precisão semântica é fundamental.

Esta pipeline de pré-processamento garante que os textos analisados são linguisticamente consistentes, semanticamente ricos e prontos para alimentar as etapas seguintes da aplicação, como a análise de palavras-chave, o agrupamento temático por clustering e a recomendação de artigos com base em similaridade textual.

3.3.2 Análise de Palavras-Chave

Uma etapa central no backend da aplicação arXplorer consiste na análise exploratória dos conteúdos textuais e da metainformação associada aos artigos científicos recolhidos. Esta análise tem como finalidade oferecer ao utilizador uma visão condensada, estruturada e informada dos principais tópicos presentes no conjunto de documentos, complementando a funcionalidade de pesquisa com indicadores agregados que orientam a interpretação e navegação dos dados. O objetivo é permitir, com um esforço cognitivo mínimo, o reconhecimento de padrões lexicais, tendências de publicação ao longo do tempo e contributos individuais por parte dos autores mais representados. Toda a lógica subjacente é executada exclusivamente no backend, e os resultados são expostos ao frontend através de endpoints dedicados.

Este router disponibiliza três funcionalidades analíticas principais:

1. Extração de palavras-chave frequentes com base na análise lexical do conteúdo textual;
2. Análise temporal da produção científica, com base nas datas de publicação dos artigos;
3. Identificação dos autores mais produtivos no conjunto de resultados obtido a partir da pesquisa.

A primeira funcionalidade consiste na extração automática dos n-gramas mais frequentes no corpora de artigos. Um n-grama é definido como uma sequência contínua de n palavras, sendo consideradas neste contexto três variantes: unigramas (termos isolados), bigramas (pares de palavras) e trigramas (sequências de três palavras). Este tipo de análise permite capturar tanto conceitos isolados como expressões compostas que ocorrem de forma recorrente no discurso científico. A metodologia adotada baseia-se na vetorização do texto por meio de uma matriz de contagem absoluta de frequências, construída com base nos títulos e resumos dos artigos. O corpus textual é previamente limpo e normalizado por meio das técnicas de pré-processamento, que incluem tokenização, remoção de stopwords e lematização, descritas na secção anterior. A decisão de utilizar contagem absoluta — em detrimento de métricas ponderadas como o TF-IDF — justifica-se pelo objetivo desta etapa e pela sua associação a uma componente exploratória que visa sintetizar os temas mais presentes no total de documentos obtidos.

A segunda funcionalidade permite analisar a evolução temporal das publicações, permitindo identificar tendências na produção científica. Para tal, calcula-se a frequência de publicações por ano a partir da data de publicação dos artigos. Esta informação permite observar o crescimento ou declínio de interesse em determinadas áreas, bem como identificar momentos de maior atividade científica relacionados com a(s) temática(s) pesquisada(s). Este tipo de análise contextualiza os tópicos extraídos, ligando o conteúdo textual à sua distribuição cronológica.

A terceira funcionalidade foca-se na análise de autores, a partir da metainformação disponível nos registos dos artigos. Cada artigo inclui a lista de autores responsáveis, e esta informação é desagregada de modo a contabilizar a frequência com que cada autor surge no conjunto de resultados. O objetivo desta análise é destacar os investigadores com maior número de publicações relevantes para a pesquisa em causa, permitindo ao utilizador identificar nomes influentes numa área específica, que poderão servir como pontos de partida para aprofundar a investigação sobre determinado tópico.

Estas três funcionalidades são executadas de forma articulada no backend, reunidas num único router responsável pela exploração dos dados recolhidos. O seu processamento permite transformar diretamente o conjunto de artigos obtido via

API do arXiv em informação estruturada e relevante, facilitando a posterior visualização e interpretação. Esta abordagem assegura que os dados enviados ao frontend já se encontram pré-processados e organizados, promovendo uma separação eficaz entre a camada lógica e a camada de apresentação, e contribuindo para a escalabilidade, manutenção e consistência da aplicação.

3.3.3 Agrupamento Temático (Clustering)

O agrupamento temático dos artigos científicos constitui uma das funcionalidades centrais do backend do arXplorer, permitindo ao utilizador identificar subconjuntos de documentos com conteúdos semanticamente semelhantes. Esta tarefa é resolvida através de métodos não supervisionados, sem necessidade de categorias pré-definidas.

O processo inicia-se com o pré-processamento textual dos resumos dos artigos, aplicando-se técnicas de normalização, remoção de stopwords e lematização, descritas na secção 3.3.1. Este tratamento assegura a eliminação de ruído e a consistência linguística necessária para a vetorização subsequente.

Para representar os textos numericamente, é utilizado o modelo TF-IDF (Term Frequency–Inverse Document Frequency). Esta técnica transforma cada resumo de um documento num vetor, onde cada dimensão corresponde a um termo e o valor representa a sua importância relativa. A pontuação TF-IDF de um termo t no resumo de um documento d é definida pela Equação I.

$$TF - IDF(t, d) = TF(t, d) \cdot \log\left(\frac{N}{DF(t)}\right)$$

Onde:

$TF(t, d)$ representa a frequência do termo t no resumo do documento d ;

N é o número total de documentos;

$DF(t)$ é o número de documentos onde o termo t ocorre no seu resumo.

Equação I - TF-IDF (Term Frequency-Inverse Document Frequency)

Com os documentos representados vetorialmente, procede-se à aplicação do algoritmo K-Means, utilizando a implementação da biblioteca scikit-learn. Este algoritmo procura dividir os vetores TF-IDF em k grupos distintos, minimizando a soma das distâncias quadradas entre cada ponto e o centroide do cluster a que pertence. A função objetivo é expressada pela Equação II.

$$\arg \min_{\{\mu_i\}} = \sum_{i=1}^k \sum_{x \in C_i} \|x - \mu_i\|^2$$

Onde:

C_i é o conjunto de vetores atribuídos ao cluster i ;

μ_i é o centroide do cluster i , calculado como a média dos vetores de C_i ;

x representa um vetor TF-IDF de um artigo.

Equação II - Função objetivo do K-means da biblioteca do scikit-learn a ser minimizada

Na implementação atual, o valor de k (número de clusters) é fixo, podendo ser ajustado diretamente nos parâmetros da função que se encontra no frontend. Após a execução do algoritmo, cada artigo recebe um rótulo numérico correspondente ao cluster a que foi atribuído.

O resultado deste processo é uma classificação temática dos artigos, que pode ser explorada e visualizada no frontend através de grafos ou listas organizadas. Esta funcionalidade proporciona ao utilizador uma visão estruturada do corpora recolhido, agrupando automaticamente os artigos em função da proximidade semântica dos seus conteúdos e promovendo uma navegação mais focada e informada.

3.3.4 Recomendação por Similaridade

A funcionalidade de recomendação por similaridade implementada no backend da aplicação arXplorer permite ao utilizador descobrir artigos cientificamente próximos ao resumo de um documento específico, facilitando a navegação por similaridade semântica. Esta operação é particularmente útil após a realização de uma pesquisa com um volume considerável de resultados, funcionando como um mecanismo adicional de filtragem e exploração contextual.

É importante reforçar que dado que a aplicação não recorre a qualquer sistema de armazenamento persistente, o processo de recomendação é executado dinamicamente e exclusivamente sobre o conjunto de artigos recolhido na pesquisa em curso. Após o utilizador submeter uma expressão/query de pesquisa, os artigos devolvidos pela API do arXiv são convertidos internamente num DataFrame em memória, que serve de base para todas as operações analíticas subsequentes, incluindo esta funcionalidade.

O utilizador pode selecionar um dos artigos devolvidos pela query (identificado de forma única através do seu Url) como ponto de partida. O backend aplica então à totalidade dos artigos do DataFrame o mesmo pré-processamento textual descrito na secção 3.3.1, normalizando, limpando e lematizando os resumos. Com base nesses textos processados, é gerada uma representação vetorial com recurso ao TF-IDF, que transforma cada resumo num vetor ponderado.

Com os vetores TF-IDF obtidos, procede-se ao cálculo da Similaridade do Cosseno entre o artigo de referência e os restantes documentos do corpora. Esta métrica mede o grau de alinhamento entre dois vetores, sendo ideal para comparar documentos de diferentes tamanhos sem que a magnitude dos vetores influencie o resultado. A fórmula é apresentada na Equação III.

$$Sim_{cos}(\vec{A}, \vec{B}) = \frac{\vec{A} \cdot \vec{B}}{\|\vec{A}\| \cdot \|\vec{B}\|}$$

Onde:

\vec{A} representa o vetor TF-IDF do artigo de referência;

\vec{B} é o vetor de outro artigo do conjunto;

O numerador é o produto interno dos dois vetores;

O denominador é o produto das suas normas (ou comprimentos).

Equação III - Similaridade do Cosseno

O backend ordena todos os artigos por grau de similaridade com o artigo selecionado e devolve os mais próximos (por omissão, os cinco mais semelhantes). O resultado inclui o título, o URL e o valor da pontuação de similaridade, permitindo que o utilizador aceda diretamente às recomendações com um clique.

Este mecanismo de recomendação é inteiramente reativo à query efetuada e funciona dentro do universo restrito dos artigos recuperados nessa pesquisa. Esta abordagem evita a necessidade de bases de dados persistentes e garante total consistência entre o contexto da pesquisa inicial e os conteúdos sugeridos, privilegiando a proximidade semântica local sobre a generalização global. Além disso, a execução em memória permite respostas rápidas e integradas ao fluxo de utilização da aplicação.

3.4 Frontend da Aplicação

O frontend do arXplorer foi desenvolvido com recurso ao framework React, uma tecnologia baseada em JavaScript amplamente utilizada para construção de

interfaces interativas e reativas. A escolha por esta tecnologia justifica-se pela sua capacidade de estruturar aplicações como composições modulares de componentes reutilizáveis, bem como pela extensa comunidade, ecossistema de bibliotecas complementares e integração nativa com o paradigma SPA (Single Page Application).

A aplicação foi desenvolvida segundo uma arquitetura modular e orientada a componentes, em que cada página corresponde a um componente isolado, localizado no diretório `./src/pages/`. Estes componentes comunicam entre si através de props e gerem o seu estado interno utilizando React Hooks, como `useState`, `useEffect` e `useCallback`. Esta abordagem funcional elimina a necessidade de classes e torna o fluxo de dados mais previsível, promovendo a legibilidade e manutenção do código.

A navegação entre as páginas da aplicação é gerida através do React Router, que permite criar uma SPA sem recarregamento de páginas, garantindo uma transição fluida entre as secções: Search, Analyze, Match e Cluster. Cada rota está mapeada para um componente React específico, o que facilita a divisão lógica da aplicação e melhora a organização do código-fonte.

Um dos principais pilares do frontend reside na sua capacidade de comunicar de forma eficiente com a lógica de processamento implementada no backend (ver Capítulo 3.3). Todas as interações com o backend são feitas por meio de uma API RESTful, em que o BASE URL é definido no módulo `./src/services/api.js`. Este módulo centraliza todas as chamadas HTTP (`fetch`) e define funções assíncronas para cada tipo de operação:

- **fetchArxivArticles** para recuperar artigos com base em palavras-chave;
- **fetchKeywords** para obter os n-gramas mais frequentes;
- **fetchTrends** para calcular tendências temporais;
- **fetchAuthors** para identificar os autores mais produtivos;
- **clusterArticles** para obter os agrupamentos temáticos;
- **fetchSimilarArticles** para gerar recomendações.

Esta abordagem centralizada e reutilizável permite isolar completamente a camada de comunicação com o backend da lógica de apresentação. Além disso, cada chamada trata internamente situações de erro, tempos de espera e mensagens de exceção.

A aplicação gera o estado local de cada componente com `useState`, ajustando dinamicamente os dados exibidos consoante a interação do utilizador. Parâmetros como o número de artigos a obter, os critérios de ordenação, ou o número de palavras-chave/autores a apresentar são controlados em tempo real, sem

recarregamento da página. Esta abordagem permite uma experiência contínua e responsiva, essencial para o tipo de exploração textual oferecido.

Por outro lado, `useCallback` e `useEffect` são utilizados para otimizar o desempenho, evitando recomputações desnecessárias e assegurando que os efeitos colaterais (como chamadas à API) ocorrem apenas quando os parâmetros de entrada são alterados.

Todos os componentes visuais foram estilizados com CSS modular, utilizando folhas de estilo específicas por página (.css). Esta abordagem garante encapsulamento, reutilização de estilos e facilidade de manutenção. A aplicação segue ainda uma filosofia responsiva, assegurando compatibilidade com dispositivos móveis, tablets e diferentes tamanhos de ecrã. Toda a interface foi testada com viewports reduzidas, garantindo uma experiência fluida em qualquer contexto de visualização.

Adicionalmente, o frontend inclui mecanismos de feedback visual, como carregadores durante o processamento (“Processing...”), mensagens de erro claras em caso de falha na comunicação com a API, e estados de vazio que informam o utilizador quando não existem resultados a apresentar. Estes elementos aumentam a robustez da interface e reforçam a clareza da interação com o sistema.

4. Resultados e Discussão

4.1 Demonstração funcional da UI

A estrutura do frontend encontra-se organizada de acordo com as funcionalidades descritas na secção 3.3, sendo cada uma delas implementada como uma página independente no diretório `./src/pages/`.

Home: ponto de entrada da aplicação, onde o utilizador pode aceder rapidamente às restantes funcionalidades. Serve como ecrã de boas-vindas e centraliza a navegação. A Figura II mostra o aspetto visual da página Home, com destaque para os botões de acesso às secções de pesquisa, análise e recomendação.

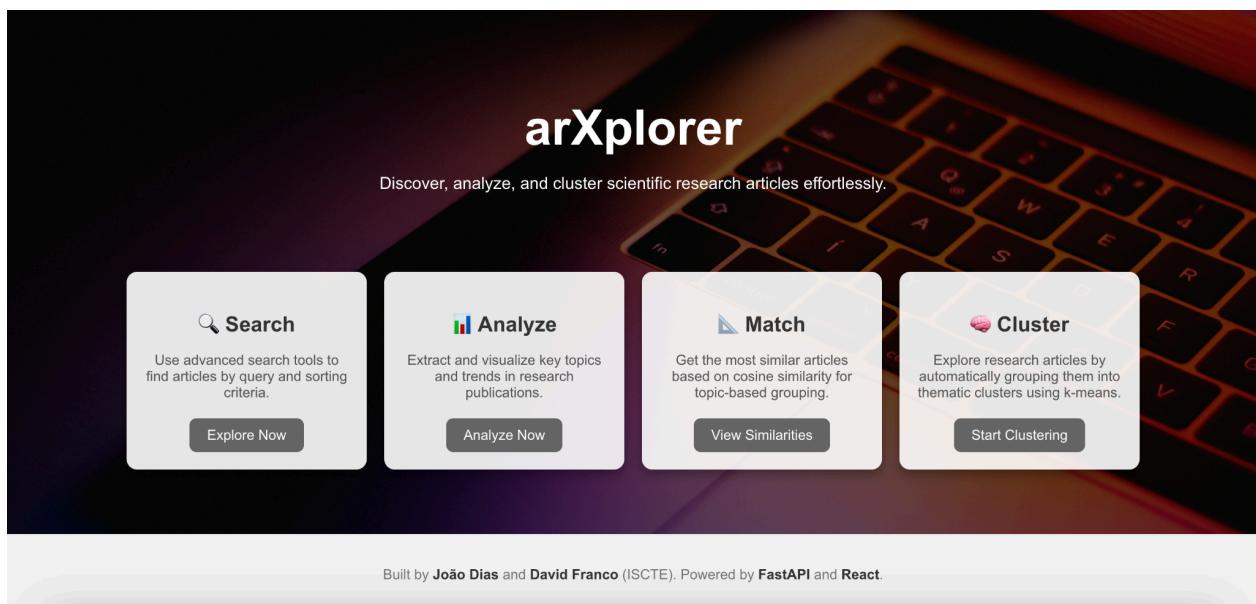


Figura II - Página Home do arXplorer

Arxiv Search: interface dedicada à pesquisa de artigos científicos na plataforma arXiv. Permite introduzir uma query textual, selecionar o número máximo de artigos e o critério de ordenação. Os resultados são apresentados com os respetivos títulos, autores, datas e resumos, organizados em páginas com navegação simples. A Figura III apresenta a UI após a submissão de uma query de pesquisa.

Arxiv Article Explorer

Search and explore research articles from Arxiv.

Search Parameters

5
Relevance
Search

Search Results

Title	Authors	Published	Summary	Link
Typesafe Modeling in Text Mining	Fabian Steeg	2011-07-28T17:46:20Z	Based on the concept of annotation-based agents, this report introduces tools and a formal notation for defining and running text mining experiments using a statically typed domain-specific language embedded in Scala. Using machine learning for classification as an example, the framework is used to develop and document text mining experiments, and to show how the concept of generic, typesafe annotation corresponds to a general information model that goes beyond text processing.	View
Very Large Language Model as a Unified Methodology of Text Mining	Meng Jiang	2022-12-19T06:52:13Z	Text data mining is the process of deriving essential information from language text. Typical text mining tasks include text categorization, text clustering, topic modeling, information extraction, and text summarization. Various data sets are collected and various algorithms are designed for the different types of tasks. In this paper, I present a blue sky idea that very large language model (VLLM) will become an effective unified methodology of text mining. I discuss at least three advantages of this new methodology against conventional methods. Finally I discuss the challenges in the design and development of VLLM techniques for text mining.	View
Pbm: A new dataset for blog mining	Mehwish Aziz, Muhammad Rafi	2012-01-10T15:18:38Z	Text mining is becoming vital as Web 2.0 offers collaborative content creation and sharing. Now Researchers have growing interest in text mining methods for discovering knowledge. Text mining researchers come from variety of areas like: Natural Language Processing, Computational Linguistic, Machine Learning, and Statistics. A typical text mining application involves preprocessing of text, stemming and lemmatization, tagging and annotation, deriving knowledge patterns, evaluating and interpreting the results. There are numerous approaches for performing text mining tasks, like: clustering, categorization, sentimental analysis, and summarization. There is a growing need to standardize the evaluation of these tasks. One major component of establishing standardization is to provide standard datasets for these tasks. Although there are various standard datasets available for traditional text mining tasks, but there are very few and expensive datasets for blog-mining task. Blogs, a new genre in web 2.0 is a digital diary of web user which has chronological entries and contains a lot of	View

[Go to Homepage](#)

Figura III - Página Arxiv Article Explorer do arXplorer

Keyword Analysis: página responsável por apresentar a análise lexical e estatística dos artigos obtidos. Exibe as palavras-chave mais frequentes sob a forma de n-gramas (unigramas, bigramas e trigramas), a distribuição temporal das publicações ao longo dos anos (gráfico de barras) e os autores mais representados no conjunto de resultados (tabela ordenada por número de publicações). Adicionalmente, apresenta a distribuição das categorias principais atribuídas a cada artigo na plataforma arXiv, informação essa que é obtida diretamente da API do arXiv durante o processo de extração dos dados. Estas categorias representam áreas científicas como cs.AI, math.ST ou physics.optics, e são agregadas num gráfico circular que sintetiza as temáticas predominantes no conjunto analisado. A Figura IV ilustra esta vista analítica, permitindo uma leitura rápida das principais tendências, temas dominantes e áreas científicas mais representadas dos 3000 artigos mais relevantes sobre Text Mining ou Machine Learning, a título ilustrativo.



Figura IV - Página Keyword Analysis do arXplorer

Similar Articles: página de recomendação baseada em similaridade semântica. O utilizador pode selecionar um artigo e visualizar os mais semelhantes, ordenados por grau de proximidade com base na métrica do cosseno. A Figura V apresenta a interface com as recomendações geradas para um exemplo concreto.

The screenshot shows the 'Find Similar Articles' interface. At the top, there is a search bar with the query "machine learning" OR "text mining", a dropdown for 'Relevance', and a 'Search Articles' button. Below the search bar, a message says "Click on an article to select it for similarity analysis." A table lists four articles:

Title	Authors	Date	Action
Meta Fine-Tuning Neural Language Models for Multi-Domain Text Mining	Chengyu Wang, Minghui Qiu, Jun Huang, Xiaofeng He	29/03/2020	View
<input checked="" type="checkbox"/> Conceptualized Representation Learning for Chinese Biomedical Text Mining	Ningyu Zhang, Qianghuai Jia, Kangping Yin, Liang Dong, Feng Gao, Nengwei Hua	25/08/2020	View
PubSqueezer: A Text-Mining Web Tool to Transform Unstructured Documents into Structured Data	Alberto Calderone	05/11/2020	View
Decoding the Alphabet Soup of Degrees in the United States Postsecondary Education System Through Hybrid Method: Database and	Sahar Vogheli, James Byars, John A Miller, Khaled Rasheed, Hamid A Arabnia	06/09/2023	View

Below the table, a section titled "Similar Articles" shows the selected article: "Conceptualized Representation Learning for Chinese Biomedical Text Mining". It includes a dropdown for "Number of Similar Articles" set to 5, and a table of five recommended articles:

Title	Similarity Score	Action
BioBERT: a pre-trained biomedical language representation model for biomedical text mining	0.52	View
BERT-based Ranking for Biomedical Entity Normalization	0.33	View
RuBioRoBERTa: a pre-trained biomedical language model for Russian language biomedical text mining	0.28	View
BioMamba: A Pre-trained Biomedical Language Representation Model Leveraging Mamba	0.26	View
Bioformer: an efficient transformer language model for biomedical text mining	0.23	View

At the bottom, there is a "Go to Homepage" button.

Figura V - Página Find Similar Articles do arXplorer

Clusters Visualization: página que permite visualizar os agrupamentos temáticos resultantes da análise de clustering. Cada artigo é mapeado para um cluster e apresentado numa visualização interativa, com suporte a zoom e drag-click, destacando relações semânticas entre artigos. É importante reforçar que são sempre apresentados 5 clusters/tópicos de interesse computados a partir dos 100 artigos mais relevantes para a query inserida pelo utilizador. A Figura VI mostra um exemplo, com a query de "machine learning", da distribuição visual dos clusters gerados.

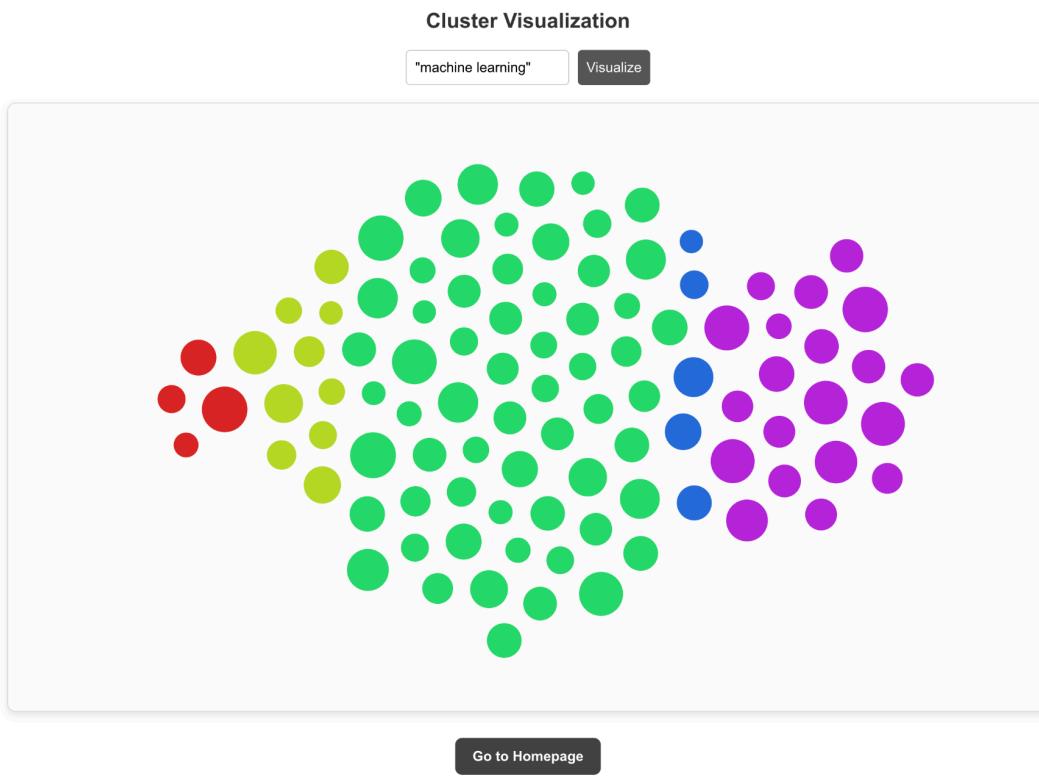


Figura VI - Página Cluster Visualization do arXplorer

Para a visualização de dados, a aplicação utiliza principalmente a biblioteca Recharts, que oferece componentes interativos. Esta biblioteca foi utilizada para representar a frequência anual de publicações (gráficos de barras), a distribuição das categorias principais dos artigos (gráficos circulares) e os n-gramas mais frequentes (listas ordenadas com contagens absolutas), permitindo ao utilizador obter rapidamente uma visão agregada do corpora de artigos analisado.

No caso específico da visualização dos agrupamentos temáticos, resultantes da aplicação do algoritmo de clustering K-Means, a aplicação recorre à biblioteca D3.js, que permite construir representações baseadas em simulações físicas. Cada artigo é representado por um círculo cuja cor corresponde ao cluster temático atribuído. A posição dos círculos é determinada dinamicamente através de uma simulação de forças, que aplica um conjunto de forças artificiais (atração para o centro, separação por cluster, colisão entre nós e distribuição horizontal) para organizar os artigos no espaço de forma fluida e informativa. Este tipo de layout melhora a legibilidade das relações semânticas e evidencia visualmente a separação entre tópicos.

4.2 Deployment

A aplicação arXplorer foi publicada integralmente na web, seguindo uma arquitetura de deployment separada entre frontend e backend. Esta separação permite uma gestão independente das camadas de apresentação e de lógica da aplicação, mantendo a coerência modular definida na arquitetura do sistema (ver Secção 3.2).

O frontend, desenvolvido com o framework React, foi alojado na plataforma Vercel, que permite o deployment contínuo a partir do repositório GitHub. Esta solução assegura a atualização automática da interface sempre que o código é alterado, e disponibiliza o serviço com elevado desempenho e suporte para aplicações single-page. A aplicação encontra-se acessível publicamente no seguinte endereço: <https://arxplorer.vercel.app>.

O backend, construído com recurso a FastAPI, foi publicado na plataforma Render. O servidor é automaticamente lançado a partir do código-fonte, com instalação das dependências e execução do módulo principal. A API gerida por esta instância expõe todas as funcionalidades da aplicação, incluindo a pesquisa, análise lexical, agrupamento temático e recomendação por similaridade.

No seu conjunto, esta abordagem garante que a aplicação está acessível a qualquer utilizador com ligação à internet, sem necessidade de instalação local, mantendo uma infraestrutura funcional e sustentável para os objetivos definidos no âmbito do projeto.

4.3 Limitações e Melhorias Futuras

Apesar das funcionalidades implementadas e da eficácia demonstrada na análise exploratória de artigos científicos, a aplicação arXplorer apresenta um conjunto de limitações que importa reconhecer. Estas restrições decorrem, em grande parte, das decisões técnicas e operacionais adotadas, bem como das próprias características dos dados e serviços utilizados.

Em primeiro lugar, a ausência de uma base de dados persistente significa que todas as operações são realizadas exclusivamente sobre o conjunto de artigos obtido a partir de uma única query, o que implica que a recomendação por similaridade e o agrupamento temático operam de forma isolada em cada sessão, sem possibilidade de comparação entre pesquisas distintas nem de armazenamento histórico. Embora esta abordagem simplifique a arquitetura e elimine dependências adicionais, limita a escalabilidade analítica da aplicação e impede análises longitudinais ou transversais a múltiplos tópicos.

Adicionalmente, o conteúdo dos artigos disponibilizado pela API pública do arXiv é limitado essencialmente a metadados, títulos e resumos. A ausência do texto completo restringe a profundidade da análise semântica, uma vez que muitos conceitos relevantes para o cálculo de similaridade ou a apresentação dos n-gramas podem estar presentes apenas no corpo do artigo. Esta limitação é particularmente relevante em domínios científicos muito densos ou técnicos, onde os resumos tendem a ser muito genéricos.

No que respeita à infraestrutura de deployment, a utilização da plataforma Render em regime gratuito impõe restrições adicionais. O backend é automaticamente suspenso após 15 minutos de inatividade, o que implica um tempo de espera inicial (cold start) de aproximadamente 60 segundos na primeira interação após um período de inatividade. Embora este atraso não comprometa a funcionalidade da aplicação, afeta negativamente a fluidez da experiência para utilizadores esporádicos ou em testes pontuais.

Por fim, o processo de agrupamento com o algoritmo K-Means assume, por simplicidade, um número fixo de 5 clusters. Esta escolha não é necessariamente ótima para todos os conjuntos de dados e pode conduzir à sub-representação ou sobreposição de tópicos, especialmente em pesquisas mais heterogéneas. A ausência de um mecanismo automático de validação (como a análise do coeficiente de Silhouette) constitui uma limitação metodológica reconhecida.

Estas limitações não invalidam a utilidade da aplicação, mas apontam caminhos claros para melhorias futuras, nomeadamente a inclusão de uma base de dados, acesso a texto completo, mecanismos de adaptação automática do número de clusters e melhoria da infraestrutura do projeto.

4.4 Comparação com Ferramentas Existentes

Atualmente, existem várias ferramentas que procuram facilitar o acesso e a análise de literatura científica, especialmente em áreas como a inteligência artificial e o processamento de linguagem natural. Entre as mais conhecidas encontram-se o Semantic Scholar, o Connected Papers e o NLP-K. Ainda que partilhem objetivos semelhantes, estas soluções distinguem-se pelas funcionalidades específicas que oferecem – e é precisamente nesse ponto que a nossa aplicação se diferencia.

O Semantic Scholar (*Lo et al., 2020*) é provavelmente a ferramenta mais consolidada e abrangente nesta área. Combina técnicas avançadas de NLP para permitir pesquisas semânticas, extração automática de entidades e geração de resumos. No entanto, apesar da sua sofisticação, funciona sobretudo como uma plataforma de consulta e agregação de informação, não oferecendo mecanismos interativos para visualização de agrupamentos temáticos nem funcionalidades

explícitas de análise lexical como n-gramas ou frequência de autores por tópico de pesquisa.

Por outro lado, o Connected Papers (n.d.) destaca-se pela capacidade de mostrar, de forma visual, as relações entre artigos com base nas suas citações. Esta rede de proximidade entre trabalhos oferece uma perspetiva interessante sobre a evolução de uma área científica. No entanto, o foco está inteiramente na estrutura bibliográfica e não no conteúdo dos artigos – o que impede análises semânticas mais profundas ou a extração de tópicos latentes diretamente a partir do texto.

Projetos mais recentes como o NLP-KG (Schopf & Matthes, 2024) apostam em arquiteturas baseadas em grafos de conhecimento para explorar relações entre conceitos e campos científicos. O NLP-KG oferece uma experiência enriquecida com funcionalidades como chatbots especializados e gráficos hierárquicos por área de estudo. No entanto, trata-se de um sistema especializado em NLP e que exige familiaridade com os conceitos técnicos da área, sendo menos acessível a utilizadores generalistas.

Neste contexto, a proposta desenvolvida no âmbito deste projeto – a aplicação arXplorer – destaca-se por integrar, de forma coesa e acessível, várias funcionalidades essenciais para quem pretende explorar rapidamente um conjunto temático de artigos: desde a pesquisa simples até à análise semântica, visualização de clusters semânticos e recomendação de artigos semelhantes. A interatividade, a clareza da interface e a abordagem modular tornam-na especialmente apelativa para utilizadores que pretendem uma exploração orientada por dados sem sacrificar a usabilidade.

5. Conclusão

A aplicação arXplorer foi desenvolvida com o objetivo de facilitar o acesso e a análise de literatura científica, especialmente para quem procura explorar novas áreas sem ter de recorrer a ferramentas complexas ou pouco intuitivas. Combinando pesquisa no arXiv, análise semântica, visualização de clusters e recomendação de artigos semelhantes, oferece uma experiência simples, mas completa, de descoberta de conhecimento.

Comparando com outras plataformas existentes, a arXplorer distingue-se por ser mais leve, acessível e orientada para a exploração visual e contextual. Apesar de ter espaço para evoluir — nomeadamente com integração de novas fontes ou técnicas mais avançadas —, cumpre bem a missão para a qual foi criada: ajudar

qualquer pessoa a perceber rapidamente os temas, tendências e autores de uma área científica.

No fundo, é uma ferramenta pensada para apoiar a curiosidade e a aprendizagem, tornando mais fácil navegar num universo de informação cada vez maior.

Referências

Bornmann, L., & Mutz, R. (2015). Growth rates of modern science: A bibliometric analysis based on the number of publications and cited references. *Journal of the Association for Information Science and Technology*, 66(11), 2215–2222.
<https://doi.org/10.1002/asi.23329>

Dessì, D., Osborne, F., Reforgiato Recupero, D., Buscaldi, D., & Motta, E. (2021). Generating knowledge graphs by employing natural language processing and machine learning techniques within the scholarly domain. *Future Generation Computer Systems*, 116, 253–264.
<https://doi.org/10.1016/j.future.2020.10.026>

Jain, A. K. (2010). Data clustering: 50 years beyond K-means. *Pattern Recognition Letters*, 31(8), 651–666.
<https://doi.org/10.1016/j.patrec.2009.09.011>

Schopf, T., & Matthes, F. (2024). NLP-KG: A System for Exploratory Search of Scientific Literature in Natural Language Processing. *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, 127–135. Association for Computational Linguistics.
<https://doi.org/10.18653/v1/2024.acl-demos.13>

Young, T., Hazarika, D., Poria, S., & Cambria, E. (2018). Recent trends in deep learning based natural language processing. *IEEE Computational Intelligence Magazine*, 13(3), 55–75.
<https://doi.org/10.1109/MCI.2018.2840738>

Kowsari, K., Jafari Meimandi, K., Heidarysafa, M., Mendu, S., Barnes, L., & Brown, D. (2019). Text classification algorithms: A survey. *Information*, 10(4), 150.
<https://doi.org/10.3390/info10040150>

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). *BERT: Pre-training of deep bidirectional transformers for language understanding*. In *Proceedings of NAACL-HLT 2019* (pp. 4171–4186).
<https://doi.org/10.48550/arXiv.1810.04805>

Beltagy, I., Lo, K., & Cohan, A. (2019). *SciBERT: A pretrained language model for scientific text*. arXiv.
<https://doi.org/10.48550/arXiv.1903.10676>

Pham, C. M., Hoyle, A., Sun, S., Resnik, P., & Iyyer, M. (2024). TopicGPT: A prompt-based topic modeling framework. In *Proceedings of NAACL 2024*.

<https://doi.org/10.18653/v1/2024.nacl-long.164>

Hogan, A., Blomqvist, E., Cochez, M., D'amato, C., de Melo, G., Gutiérrez, C., Kirrane, S., Labra Gayo, J. E., Navigli, R., Neumaier, S., Ngonga Ngomo, A.-C., Polleres, A., Rashid, S. M., Rula, A., Schmelzeisen, L., Sequeda, J., Staab, S., Zimmermann, A., & d'Amato, C. (2021). Knowledge graphs. *ACM Computing Surveys*, 54(4), 1–37.

<https://doi.org/10.1145/3447772>

Lo, K., Wang, L. L., Neumann, M., Kinney, R., & Weld, D. S. (2020). S2ORC: The Semantic Scholar Open Research Corpus. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (pp. 4969–4983).

<https://doi.org/10.18653/v1/2020.acl-main.447>

Batinica, B., & Treleaven, P. C. (2015). Social media analytics: A survey of techniques, tools and platforms. *AI & Society*, 30(1), 89–116.

<https://doi.org/10.1007/s00146-014-0549-4>

Adhya, S., & Sanyal, D. K. (2024). GINopic: Topic Modeling with Graph Isomorphism Network. Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), 6171–6183.

<https://doi.org/10.18653/v1/2024.nacl-long.342>

Wang, H., Prakash, N., Hoang, N. K., Hee, M. S., Naseem, U., & Lee, R. K.-W. (2023). Prompting large language models for topic modeling. *arXiv*.

<https://arxiv.org/abs/2312.09693>

Cvejoski, K., Sánchez, R. J., & Ojeda, C. (2023). Neural Dynamic Focused Topic Model. Proceedings of the AAAI Conference on Artificial Intelligence, 37(11), 12719–12727.

<https://doi.org/10.1609/aaai.v37i11.26496>

Grootendorst, M. (2022). BERTopic: Neural topic modeling with a class-based TF-IDF procedure. *arXiv*.

<https://doi.org/10.48550/arXiv.2203.05794>

Connected Papers. (n.d.).

<https://www.connectedpapers.com>

Schopf, T., & Matthes, F. (2024). *NLP-KG: A System for Exploratory Search of Scientific Literature in Natural Language Processing*. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)* (pp. 127–135). Association for Computational Linguistics.
<https://doi.org/10.18653/v1/2024.acl-demos.13>