Sistema Assistido para Seleção de RH (SA4S-RH)

Universidade Fernando Pessoa

Samuel Marques Mota

Orientador: Prof. Doutor Rui Silva Moreira

Co-orientador: Prof. Doutor Ivo Pereira

Universidade Fernando Pessoa

Tese submetida para obtenção do grau de

Mestre em Engenharia Informática, ramo Computação Móvel

2023/2024

Resumo

A complexidade envolvida na seleção de candidatos pelas empresas exige a aplicação de diversas tecnologias, que desempenham um papel crucial na otimização da triagem e na escolha dos profissionais mais adequados para cada posição. As metodologias tradicionais apoiam-se sobretudo na correspondência direta (*matching*) de palavras-chave (*keywords*) relevantes extraídas do currículo e da descrição da vaga. Contudo, estas limitam-se à avaliação superficial, sendo pouco adaptativas devido à grande variação semântica da linguagem. Para resolver esse problema foi proposta a *pipeline* SA4S-RH, onde foram usadas ferramentas com base na análise de processamento de linguagem natural, para analisar e interpretar o contexto das *skills* apresentadas de cada candidato com base na análise textual profunda e a atribuição de uma *skill* na taxonomia ESCO com recurso a um modelo de linguagem (*Large Language Model*). Esta abordagem permitiu a normalização das competências extraídas de cada candidato e dos requisitos da descrição da vaga para poder ser feita uma comparação justa. Paralelamente, esta abordagem permite tirar conclusões sobre o estado atual do mercado de trabalho, como as competências mais procuradas e as mais em falta.

Para avaliar os resultados obtidos pelo modelo de linguagem (*Large Language Model*) na tarefa de atribuição de entidades (*Entity Linking*) da taxonomia ESCO, foram definidas métricas claras e mensuráveis, como a taxa de acerto em relação a um *dataset* sintético, anotado com a entidade ESCO esperada e a atribuída pelo modelo. Essa abordagem permitiu uma avaliação objetiva do desempenho da *pipeline* SA4S-RH, com foco na precisão do modelo na correspondência entre as menções de *Skills* extraídas e as entidades da taxonomia ESCO.

Os resultados obtidos pela aplicação da *pipeline* SA4S-RH mostraram uma promissora capacidade na utilização de LLM para tarefas de processamento de linguagem natural, na tarefa de *Entity Linking*. A precisão de 70,63%, alcançada no reconhecimento correto de *Skills* indica que a utilização de LLMs para tarefas de *Entity Linking* pode ser eficaz em contextos onde a identificação de habilidades profissionais a partir de descrições textuais é crítica, bem como na análise de vagas de emprego ou currículos.

Gostaria de dedicar esta tese à minha família, pelo apoio incondicional e encorajamento ao longo de todo o meu percurso académico. Agradeço também aos meus orientadores, pela orientação, paciência e pelos valiosos conhecimentos partilhados, que foram fundamentais no desenvolvimento deste trabalho.

Agradecimentos

A realização desta tese não teria sido possível sem o apoio e encorajamento de várias pessoas e instituições, às quais expresso os meus sinceros agradecimentos.

Primeiramente, gostaria de agradecer aos meu orientadores **Prof. Doutor Rui Silva Moreira** e **Prof. Doutor Ivo Pereira** , pela sua orientação, paciência e apoio contínuo durante todo o processo. As suas valiosas sugestões e conhecimentos especializados foram essenciais para o desenvolvimento deste trabalho.

Agradeço também à minha família e amigos, pelo apoio incondicional e constante motivação ao longo desta jornada. O equilíbrio que me ajudaram a manter foi essencial, assim como a compreensão e paciência diante do tempo que tive de dispensar. Sem a compreensão, o suporte emocional e encorajamento, a conclusão deste projeto teria sido muito mais difícil.

Acrónimos

ATS *Applicant Tracking System*

ESCO *European Skills, Competences, Qualifications and Occupations*

NLP *Natural Language Processing*

EL *Entity Linking*

LLM *Large Language Models*

EU *European Union*

CV *Curriculum Vitae*

NER *Named Entity Recognition*

ML *Machine Learning*

DNN *Deep Neural Network*

CNN *Convolutional neural network*

RNN *Recurrent Neural Network*

AI *Artificial Inteligence*

NN *Neural Networks*

FFNN *Feed-Forward Neural Network*

LM *Language Models*

SLM *Statistical Language Models*

MLM *Masked Language Model*

MLM *Masked Language Modeling*

PLM *Pre-trained Language Models*

NLM *Neural Language Models*

DL *Deep Learning*

GPT *Generative Pre-trained Transformers*

IR *Information Retrieval*

DRMM *Deep Relevance Matching Model*

SNRM PRF *Standalone Neural Ranking Model with Pseudo-Relevance Feedback*

GRU *Gated Recurrent Unit*

LSTM *Long Short Term Memory network*

DBN *Deep Belief Network*

NMT *Neural Machine Translation*

KB *Knowledge Base*

XMLC *Extreme Multilabel Classification*

GAN *Generative adversarial network*

PICO *Patient or Population, type of Interventions, Comparisons and Outcomes*

PRISMA *Preferred Reporting Items for Systematic reviews and Meta-Analyses*

FAISS *Facebook AI Similarity Search*

GPU *Graphics Processing Unit*

RAG *Retrieval-Augmented Generation*

RAM *Random-access Memory*

CPU *Central Processing Unit*

ERP *ESCO Relation Prediction*

API *Application Programming Interface*

SA4S-RH *Sistema Assistido para Seleção de Recursos Humanos*

ZSL *Zero-shot learning*

FSL *Few-Shot Learning*

Índice

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

1.1 Contexto e Problema

A crescente complexidade na seleção de candidatos por parte das empresas, devido à emergente criação de trabalhos especializados que requerem múltiplas competências em diversas areas e tecnologias e à acessível abertura a candidaturas digitais por parte dos candidatos, leva à necessidade de utilizar diversas tecnologias que permitam auxiliar no processo seletivo por parte dos recursos humanos para escolher, entre vastas quantidades de candidatos, os mais compatíveis com o requerido para cada vaga.

Metodologias tradicionais de seleção de candidatos, como os sistemas de *Applicant Tracking System* (ATS) apoiam-se sobretudo no processo de correspondência direta (*matching*) de palavras-chave relevantes extraídas do currículo e da descrição da vaga de emprego. Contudo, esta abordagem limita-se à avaliação superficial, mostrando baixa adaptabilidade às variações linguísticas e contextuais, bem como às nuances presentes nesses cenários (Zhang, 2024).

1.2 Objetivos

Este trabalho propõe-se a colmatar a lacuna identificada na tese de doutoramento de (Zhang, 2024), quanto à atribuição de uma *Skill* padronizada na taxonomia *European Skills, Competences, Qualifications and Occupations* (ESCO) a uma *Skill* extraída do texto dos currículos e das descrições das vagas. Esta tarefa de *Natural Language Processing* (NLP) denominada de atribuição de entidades *Entity Linking* (EL) é especialmente importante pois permite padronizar e facilitar a correspondência entre currículos e vagas, e tirar informações valiosas sobre as exigências do mercado de trabalho (Zhang, 2024). Para isso, utilizou-se as capacidades dos *Large Language Models* (LLM) para melhorar a eficácia do sistema de atribuição de entidades ESCO, aproveitando a capacidade de raciocínio destes modelos para decidir a entidade que melhor se adequa a cada *Skill* extraída.

Este método inovador visa melhorar a precisão e a relevância das correspondências entre as competências dos candidatos e os requisitos das vagas, superando as limitações das abordagens tradicionais baseadas apenas na correspondência de palavras-chave. Ao utilizar LLMs espera-se não só capturar as nuances contextuais das *Skills* descritas, mas também proporcionar uma análise mais sofisticada e adaptativa, alinhada com as necessidades reais do mercado de trabalho.

Além disso, a integração da taxonomia ESCO, uma classificação multilíngua que identifica e categoriza aptidões, competências, qualificações e profissões relevantes para o mercado de trabalho e educação na *European Union* (EU), desenvolvida pela Comissão Europeia desde 2010, facilita a normalização e interoperabilidade dos dados de competências. Sendo uma estrutura padronizada e amplamente reconhecida, a ESCO contribui para maior consistência e comparabilidade nos processos de seleção de candidatos. Esta normalização permite uma comparação justa entre as *Skills* presentes nos *Curriculum Vitae*s (CVs) e aquelas exigidas nas descrições de vagas de emprego.

Avançando o estado da arte, esta investigação foca na aplicação de LLMs em tarefas de EL, oferecendo uma abordagem prática e eficaz que aprimora os processos de recrutamento e seleção. Com isso, possibilita uma avaliação mais padronizada e justa, trazendo benefícios significativos tanto para candidatos quanto para empregadores.

Inserido no contexto do mercado de trabalho, o foco da pesquisa é a extração de *Skills* tanto dos candidatos quanto das descrições de vagas e na atribuição de entidades na taxonomia ESCO. Os modelos utilizados foram especialmente treinados e adaptados para esse cenário, visando aumentar a precisão e eficiência do processo seletivo.

A relevância desta investigação está na contribuição ao avanço do estado da arte em NLP, com especial ênfase na aplicação de LLM em tarefas de EL. Embora ainda pouco otimizados e eficientes, esses modelos têm grande potencial. A aplicação de LLMs em tarefas de atribuição de de entidades na taxonomia ESCO, pode trazer avanços significativos em tarefas de atribuição de entidades (EL), o que é crucial não apenas na seleção de candidatos, mas em várias outras áreas relacionadas ao mercado de trabalho.

É abordado também as limitações das metodologias tradicionais de correspondência de palavras-chave, propondo soluções mais avançadas e adaptativas. Os modelos utilizados neste trabalho são ajustados para melhor capturar as nuances linguísticas e contextuais presentes nos currículos e nas descrições de vagas, oferecendo uma análise mais profunda e precisa das competências dos candidatos.

Este trabalho não só se propõe a fornecer ferramentas práticas e inovadoras para aprimorar os processos de seleção de recursos humanos, contribuindo para um mercado de trabalho mais eficiente e justo, mas também a avançar o conhecimento na aplicação de LLM para EL.

Este trabalho tem também como objetivo melhorar as tecnologias tradicionais utilizadas pelos recursos humanos nos processos de seleção de candidatos, fornecendo um

sistema capaz de normalizar *Skills* de candidatos e atribuir a correspondente *Skill* da taxonomia ESCO associada. Através desta normalização é possível tirar fazer uma comparação justa.

A escolha de LLM para o processo de atribuição de *Skills* extraídas dos currículos dos candidatos a *Skills* da taxonomia ESCO prende-se às capacidades demonstradas no estado da arte dos mesmos relativas a aprendizagem por contexto e capacidade de raciocínio.

1.3 Metodologia

O sistema será composto por um modelo tradicional de NLP, treinado no contexto do mercado de trabalho para a tarefa de *Named Entity Recognition* (NER), que envolve a extração de *Skills* de texto não estruturado relativo aos currículos e da descrição da vaga.

A metodologia apresentada para o aprimoramento dos sistemas de atribuição de entidades EL com a utilização de LLMs baseou-se na criação de uma base de dados vetorial, desenvolvida a partir de vetores de *embedding* gerados com base num *dataset* que contém múltiplos exemplos textuais para cada entidade da taxonomia ESCO. Esta base de dados foi utilizada para filtrar do conjunto de entidades *Skill* da ESCO, aquelas que são mais prováveis de serem atribuídas às *Skills* extraídas do texto não estruturado, tanto de currículos como de descrições de vagas. Posteriormente, são usadas as capacidades do LLM local, desenvolvido pela *Meta*, *LLama2*, através de engenharia de *prompt* e utilizando técnicas de *in-context learning* e *zero-shot learning*, fazendo com que este selecione a entidade ESCO que melhor se adequa entre as mais prováveis previamente filtradas.

1.4 Estrutura do Documento

O presente documento está dividido em 5 capítulos. O primeiro capítulo introduz o tema do projeto onde se apresenta e retrata o problema da complexidade no processo seletivo por parte das empresa, bem como se identificam os objetivos a alcançar com o desenvolvimento do trabalho.

No capítulo 2 apresenta-se a revisão bibliográfica detalhada dos trabalhos relacionados que contribuíram diretamente para o desenvolvimento deste projeto. São introduzidos conceitos básicos que serão abordados ao longo do trabalho e exploradas as principais abordagens, tendências emergentes e conquistas que moldaram o campo nos últimos anos, assim como os desafios e limitações que ainda persistem. A análise crítica dessas pesquisas fornece uma base sólida para a elaboração da metodologia proposta, justificando a sua necessidade e destacando o potencial de aprimoramento em relação às lacunas identificadas nas abordagens existentes. O capítulo 3 introduz o sistema SA4S-RH, descrevendo de forma detalhada a arquitetura e requisitos do mesmo, além de justificar as decisões

tomadas e relatar a implementação do trabalho. O capítulo 4 é dedicado à fase de testes e análise dos resultados obtidos pelo sistema proposto, apresentando uma avaliação crítica da sua eficácia.

Por fim, no capítulo 5 encontram-se as conclusões finais deste projeto de dissertação, bem como sugestões para possíveis desenvolvimentos futuros.

Capítulo 2

Estado da Arte na Automatização da Seleção de Recursos Humanos

2.1 Background Tecnológico em Processamento de Linguagem Natural

Esta secção servirá como uma introdução e definição dos conceitos fundamentais que constituirão o *background* tecnológico necessário para o desenvolvimento deste projeto.

O processamento de linguagem natural, também conhecido como linguística computacional, é uma área da ciência da computação que se dedica ao desenvolvimento de modelos e processos computacionais para resolver problemas relacionados à compreensão automática das línguas humanas. O objetivo central da NLP é criar soluções tecnológicas capazes de processar e interpretar o significado de textos ou fala, automatizando tarefas complexas de análise e manipulação da linguagem natural (Otter et al., 2019).

Os problemas abordados em NLP podem ser agrupados em duas grandes categorias: área central (*core area*) e de aplicação (*application area*). A área central trata de questões relacionadas à modelação de linguagem, que procura quantificar as associações entre palavras em contextos naturais; o processamento morfológico, que segmenta palavras nos seus componentes significativos e determina as suas classes gramaticais; o processamento sintático responsável por construir diagramas ou árvores sintáticas das frases; e o processamento semântico, que visa compreender o significado de palavras, frases e estruturas maiores no texto (Otter et al., 2019). Na área de aplicação de NLP incluem-se temas como a extração de entidades nomeadas e relações (NER), tradução automática entre línguas, o resumo automátio, a geração de texto e a classificação textual. Estas aplicações muitas vezes requerem a integração das abordagens das áreas centrais, aplicando modelos e técnicas específicas para resolver problemas práticos de interesse comercial ou académico (Otter et al., 2019).

Nas últimas décadas, a área de NLP evoluiu significativamente de abordagens ba-

seadas em regras e estatísticas para métodos orientados por dados, utilizando técnicas de *Machine Learning* (ML) para treinar modelos. Modelos tradicionais de NLP, como *Naïve Bayes, k-nearest neighbors, hidden Markov, etc*, foram gradualmente substituídos e complementados por modelos neuronais mais sofisticados, que se têm mostrado superiores em várias tarefas de NLP (Otter et al., 2019).

2.1.1 Redes e Modelos Neuronais

Os modelos neuronais, como redes neuronais profundas (*Deep Neural Network*s (DNNs)) e redes neuronais convolucionais (*Convolutional neural network*s (CNNs)), revolucionaram o campo de NLP ao permitir que os sistemas aprendam automaticamente a partir de grandes quantidades de dados textuais, reconhecendo padrões complexos e melhorando o desempenho em tarefas de NLP (Otter et al., 2019). Paralelamente, o uso de redes neuronais recorrentes (*Recurrent Neural Network*s (RNNs)) e, mais recentemente de redes *Transformer*, ampliou a capacidade dos sistemas de NLP para lidar com a dependência temporal e relacional entre palavras em sequências de texto, tornando estes modelos mais eficientes em tarefas de NLP (Otter et al., 2019).

O rápido progresso das técnicas de NLP impulsionou a adoção dessas tecnologias numa vasta gama de aplicações práticas, desde assistentes virtuais até sistemas de análise de sentimento de mercados, mostrando o impacto transformador dos modelos de NLP no campo de *Artificial Inteligence* (AI) (Otter et al., 2019).

Uma rede neuronal (*Neural Networks* (NN)) é um tipo de modelo de *Machine Learning* (ML) baseado em conjuntos de funções matemáticas chamadas neurónios (ou nós) que calculam o valor de saída com base em alguma entrada, sendo que o poder da rede neuronal advém das conexões entre os neurónios. Cada neurónio está ligado a alguns dos seus pares e a força de cada conexão é quantificada através de um peso numérico. Estes determinam em que medida a saída de um neurónio será levada em consideração como uma entrada para um neurónio seguinte (Otter et al., 2019). Cada um destes nós nas camadas de saída realiza uma soma ponderada dos valores recebidos dos nós de entrada e em seguida gera saídas, utilizando funções de transformação não lineares simples nessas somas (Otter et al., 2019). Correções nos pesos entre os nós são feitas em resposta aos erros individuais e perdas que as redes exibem nos nós de saída, onde é geralmente utilizada a descida de gradiente estocástica, considerando as derivadas dos erros nos nós, numa abordagem chamada retropropagação.

Os principais fatores que distinguem os diferentes tipos de redes são a forma como os nós estão conectados e o número de camadas. Redes básicas, nas quais todos os nós podem ser organizados em camadas sequenciais, onde cada nó recebe entradas apenas dos nós nas camadas anteriores, são conhecidas como redes neuronais *feedforward* (*Feed-Forward Neural Network*s (FFNNs)). Embora não haja um consenso claro sobre o que

define exatamente uma rede neuronal profunda (DNN), geralmente considera-se que as redes com várias camadas ocultas (ou seja, camadas que recebem entradas da camada de entrada ou de outras camadas ocultas) são profundas e aquelas com muitas camadas são consideradas muito profundas (Otter et al., 2019). Um exemplo de arquitetura profunda são as redes neuronais convolucionais (CNNs), que se baseiam na operação de convolução matemática e no processamento de sinais. As CNNs utilizam filtros que permitem a análise simultânea de diferentes características nos dados, sendo amplamente aplicadas no processamento de imagens e vídeos, assim como em tarefas de processamento de fala e NLP (Otter et al., 2019).

2.1.2 Modelos de Linguagem

A modelação da linguagem (*Language Models* (LM)) é um aspeto fundamental do processamento da linguagem natural tendo evoluído através de várias fases, culminando no desenvolvimento de modelos estatísticos de linguagem (*Statistical Language Model*s (SLMs)), modelos de linguagem neuronal (*Neural Language Model*s (NLMs)), modelos de linguagem pré-treinados (*Pre-trained Language Model*s (PLMs)) e, mais recentemente, modelos de linguagem em grande escala (LLMs) (Minaee et al., 2024; Zhao et al., 2024).

Ao longo das décadas, os modelos estatísticos de linguagem (SLM) tornaram-se essenciais para várias tarefas de compreensão e geração de texto em linguagem natural (NLP), incluindo o reconhecimento de fala, a tradução automática e a recuperação de informação (Minaee et al., 2024; Zhao et al., 2024). Estes interpretam texto como uma sequência de palavras e estimam a probabilidade do texto como o produto das probabilidades das palavras. O tipo predominante de SLM são os modelos de cadeia de *Markov*, conhecidos como modelos *n-gram*, que calculam a probabilidade de uma palavra com base nas *n-1* palavras precedentes. Para lidar com a esparsidade dos dados (atribuição de probabilidades zero a palavras ou *n-gram* não vistos), usam-se técnicas de suavização. No entanto, esses modelos são limitados e não conseguem capturar completamente a diversidade e variabilidade da linguagem natural (Minaee et al., 2024; Zhao et al., 2024).

Os NLM lidam com a esparsidade dos dados ao mapear palavras para vetores contínuos de baixa dimensão (vetores de *embedding*), onde preveem a próxima palavra com base na agregação dos vetores de *embedding* das palavras precedentes, utilizando redes neuronais. Os vetores de *embedding* permitem calcular a similaridade semântica entre entradas, independentemente das suas formas ou modalidades. Os primeiros NLM eram específicos para tarefas, sendo treinados em dados específicos de tarefas e aprendendo espaços de *embeddings* também específicos (Minaee et al., 2024). Ao contrário dos NLM, os PLM são independentes de tarefas específicas, oferecendo maior generalização no espaço de *embeddings* aprendido. Estes seguem o paradigma de pré-treino e *finetuning*

onde são pré-treinados em grandes *Corpus* (conjunto de dados que consiste em recursos linguísticos nativamente digitais e mais antigos, digitalizados) de texto não anotado para tarefas gerais e, em seguida, ajustados para tarefas específicas com pequenas quantidades de dados anotados (Minaee et al., 2024; Otter et al., 2019).

Os LLMs são modelos de linguagem avançados, pré-treinados e de grande escala compostos por redes neuronais (NN) com uma quantidade extremamente elevada de parâmetros, que podem variar de dezenas de milhões até triliões (Zheng et al., 2023; Gan et al., 2024). Estes são desenvolvidos para entender e gerar texto em linguagem natural onde desempenham um papel crucial na capacidade de compreender e gerar texto semelhante à linguagem humana. Para capturar padrões complexos da linguagem, são treinados através de técnicas de *Deep Learning* (DL) em grandes conjuntos de dados textuais, permitindo capturar padrões complexos da linguagem (Minaee et al., 2024; Zhao et al., 2024). O processo de treino envolve grandes volumes de texto não anotados, utilizando métodos de aprendizagem auto-supervisionada ou semi-supervisionada (Zheng et al., 2023; Gan et al., 2024).

Além disso, os LLMs fazem parte do tipo de modelos de AI baseados NN, sendo compostos por milhões de neurónios interligados cada um operando como uma unidade de processamento elementar. Um LLM pode ter milhões de neurónios com muitas centenas de milhares de milhões de ligações entre eles, sendo que cada ligação tem o seu próprio peso. Assim como nas redes neutrais convencionais (CNN), a força e eficácia do LLM reside nas interconexões entre estes neurónios, cujos pesos numéricos determinam a influência relativa das saídas de neurónios precedentes nas entradas dos neurónios subsequentes (Minaee et al., 2024).

Os LLM são principalmente baseados na arquitetura *Transformers* (Wolf et al., 2020), que possuem dezenas a centenas de milhares de milhões de parâmetros, sendo pré-treinados em grandes volumes de dados textuais. Esta arquitetura é caracterizada pelo uso de mecanismos de atenção, permitindo que os modelos se adaptem a uma ampla gama de tarefas de NLP (Zheng et al., 2023; Gan et al., 2024). Comparados aos *Pre-trained Language Models* (PLM), os LLM não são apenas maiores em tamanho, mas também exibem habilidades de compreensão e geração de linguagem mais fortes e habilidades emergentes que não estão presentes em modelos de menor escala (Minaee et al., 2024; Zhao et al., 2024).

Os LLMs existentes podem ser classificados em duas categorias principais: discriminativos e generativos (Zheng et al., 2023; Wu et al., 2023). Modelos discriminativos, como o *BERT*, utilizam uma arquitetura de transformador bidirecional profundo e o objetivo de *Masked Language Modeling* (MLM) para pré-treino (Zheng et al., 2023; Skondras et al., 2023), enquanto os modelos generativos, como o *Generative Pre-trained Transformers* (GPT), focam-se em melhorar a compreensão da linguagem através de pré-treino generativo, no qual o modelo aprende a prever a próxima palavra numa frase sem qualquer tarefa específica (Zheng et al., 2023; Wu et al., 2023). A introdução de modelos

como o *GPT-3*, que possui 135 mil milhões de parâmetros, representou um salto significativo tanto em escala quanto em capacidade qualitativa, demonstrando o impacto do aumento do *corpus* de pré-treino e do número de parâmetros na eficácia dos LLMs. Para além disso os LLMs generativos têm-se mostrado particularmente eficazes na geração de texto coerente e contextualmente relevante, tendo sido aplicados com sucesso em sistemas de recomendação e outras aplicações específicas através de estratégias de adaptação, como *fine-tuning* e *instruction tuning* (Wu et al., 2023; Zheng et al., 2023; Gan et al., 2024). Estes mecanismos facilitam a adaptabilidade dos LLMs a diversas tarefas de NLP, demonstrando *performance* e efetividade relevante em relação a outros modelos de NLP mais tradicionais (Gan et al., 2024). Os LLMs não só revolucionaram a área de NLP, mas também abriram novas possibilidades em diversos domínios de aplicação, tornando-se ferramentas cruciais para o processamento e a geração de linguagem em grande escala.

No processo de desenvolvimento os LLMs são treinados a analisar grandes volumes de dados textuais. Durante o processo de pré-treino o modelo, inicialmente não treinado, possui os pesos inicializados aleatoriamente, onde é treinado para prever o próximo *token* (correspondente a palavras, conjuntos de caracteres ou combinações de palavras e pontuação que são usados por LLMs) numa sequência textual. O processo de pré-treino utiliza o método de aprendizagem auto-supervisionada, isto porque não requer anotação manual dos dados de treino. Em vez disso, as anotações são derivadas dos próprios dados durante o pré-treino. O modelo recebe sequências de *tokens* de grandes quantidades de texto onde estes aprendem a prever o próximo *token* nessas sequências. Se a previsão estiver incorreta, o erro é calculado e os pesos do modelo são ajustados para melhorar as previsões futuras (Minaee et al., 2024). Estas adaptações iterativas visam otimizar o desempenho do LLM na tarefa específica para a qual foi concebido, resultando na capacidade de compreensão e geração de texto em linguagem natural. A interconectividade complexa, aliada à capacidade de processamento sequencial inerente à arquitetura de *Transformers*, confere aos LLMs a notável eficácia na manipulação e compreensão de dados textuais, destacando a relevância e aplicabilidade em diversas aplicações de AI (Minaee et al., 2024; Zhao et al., 2024).

A arquitetura de NN, *Transformers*, é projetada para o processamento sequencial de dados, tornando-os ideais para lidar com texto. Ao contrário das NN tradicionais, os modelos baseados em *Transformers* enfatizam a "atenção", onde alguns neurónios atribuem mais peso a outros específicos dentro de uma sequência. Esta arquitetura alinha-se bem com a natureza sequencial da linguagem, melhorando as capacidades de geração de texto dos LLMs (Minaee et al., 2024; Wolf et al., 2020; Zhao et al., 2024).

Exemplos de LLM são os modelos *GPT*-3.5 e 4 (baseado na arquitetura *Generative Pre-trained Transformers*, desenvolvido pela *OpenAI*). Estes modelos são capazes de realizar varias tarefas de processamento de linguagem natural NLP, como tradução automática, geração de texto, resposta a perguntas e criação de código de programação (Minaee

et al., 2024; Zhao et al., 2024).

Os LLMs podem ser adaptados e refinados para tarefas específicas utilizando diversas técnicas intrínsecas à sua arquitetura, entre as quais:

In-context learning

Aprendizagem em Contexto (*In-context learning*) é o método de engenharia de *prompt* (*prompt engineering*) onde os LLMs aprendem uma nova tarefa a partir de um pequeno conjunto de demonstrações apresentadas na *prompt* durante a inferência, o que permite resolver diversas tarefas de NLP sem a necessidade de fazer *fine-tuning* ao modelo (Minaee et al., 2024).

Os LLMs tem a capacidade de realizar dois tipos de estruturas de aprendizagem pela configuração da *prompt*: *few-shot* e *zero-shot learning*. A técnica de *few-shot learning* permite que o LLM execute tarefas fornecendo apenas alguns exemplos no contexto da *prompt*, por meio do processo conhecido como *in-context learning*. Por outro lado, o *zero-shot learning* capacita o modelo a resolver tarefas baseando-se apenas em instruções, sem a necessidade de exemplos explícitos apresentados na *prompt*. Esse processo é conhecido como também *prompting*, onde o modelo é condicionado a gerar respostas com base em uma entrada específica (Minaee et al., 2024; Kojima et al., 2023). As capacidades de raciocínio dos LLMs são significativamente aumentadas através do uso de *fine-tuning* ou *few-shot learning*, criando raciocínio passo a passo (Kojima et al., 2023). No entanto, apenas a utilização de *zero-shot learning* evidencia melhorias substanciais à capacidade de raciocínio de LLMs em diversas tarefas de NLP, sem a necessidade de *fine-tuning* ao modelo (Minaee et al., 2024; Kojima et al., 2023). Os LLM podem resolver tarefas complexas, dividindo-as em etapas intermédias de raciocínio como demonstrado na *prompt* de cadeia de pensamento (Minaee et al., 2024).

Fine-tuning e Instruction Tuning

Os LLMs mais recentes não necessitam de ser ajustados a um conjunto de dados para serem usados num processo denominado de *fine-tuning*. No entanto, podem beneficiar de *fine-tuning* para a tarefa específica ou para os dados pretendidos (Minaee et al., 2024; Kojima et al., 2023). A principal vantagem de ajustar os LLMs é a possibilidade de alinhar as respostas às expectativas que se têm ao fornecer instruções através das *prompts*. Este tipo de *fine-tuning* é denominado de ajuste de instruções (*instruction tuning*), onde os LLMs passam a seguir instruções para novos tipos de tarefas sem ter de usar exemplos explícitos nas *prompts* (Minaee et al., 2024).

O processo de *fine-tuning* pode ser realizado para uma única ou mais tarefas, existindo diversas abordagens para *fine-tuning* em múltiplas tarefas, o que melhora os resultados e reduz a complexidade da necessidade de engenharia das *prompts*. Além disso, fazer *fine-*

tuning ao LLM é recomendável para expor o modelo a novos dados ou dados proprietários que não foram incluídos durante o pré-treino do mesmo (Minaee et al., 2024; Kojima et al., 2023).

2.1.3 Embeddings

No processamento de linguagem natural (NLP), *embeddings* desempenham um papel crucial na representação de palavras ou componentes linguísticos como vetores de valores numéricos, permitindo que os modelos de aprendizagem automática compreendam e processem a linguagem de forma mais eficaz. Estas representações vetoriais, frequentemente denominadas *word embeddings*, capturam as relações semânticas e sintáticas entre as palavras. Ao contrário das abordagens anteriores em que as palavras eram tratadas como entidades distintas, *embeddings* permitem que palavras com significados ou contextos semelhantes tenham representações vetoriais comparáveis. Isto ajuda a superar desafios como lidar com sinónimos ou palavras que não foram vistas durante o treino, conhecidas como palavras fora do vocabulário (*out-of-vocabulary words*) (Otter et al., 2019).

Embeddings são tipicamente criadas através de técnicas de modelação de linguagem, onde os modelos são treinados para prever a próxima palavra numa sequência com base nas palavras anteriores. Neste processo os estados internos das redes neuronais podem ser extraídos para formar essas representações vetoriais. As representações de *embeddings* resultantes geralmente têm entre 50 a 300 dimensões, permitindo uma codificação detalhada das relações linguísticas. Um exemplo famoso que demonstra a utilidade das *word embeddings* é a seguinte analogia: a palavra "rei" está para "rainha" assim como "homem" está para "mulher". Nesta analogia a diferença vetorial entre "rei" e "rainha" aproxima-se da diferença vetorial entre "homem" e "mulher", ilustrando assim a forma como a representação em *embeddings* captura essas analogias semânticas de forma matematicamente significativa (Otter et al., 2019).

Os modelos de linguagem baseados em redes neuronais (NN) têm impulsionado o desenvolvimento de *embeddings*, passando de modelos estatísticos simples para técnicas de DL, como os a arquitetura *Transformers*, que conseguem capturar relações mais amplas e complexas nos textos. Estes avanços tornaram os *embeddings* o formato de entrada padrão para os sistemas de NLP modernos, alimentando aplicações que vão desde a tradução automática até ao resumo de texto, onde a compreensão da interação complexa entre as palavras é essencial para gerar saídas linguísticas precisas e coerentes (Otter et al., 2019).

2.1.4 Tarefas de Processamento de Linguagem Natural

Tarefas de NLP cobrem uma vasta gama de aplicações, cada uma abordando um aspeto específico da compreensão ou manipulação da linguagem. Tarefas fundamentais da NLP, como a recuperação de informação, a extração de informação, a classificação de texto, a

geração de texto, o resumo, a resposta a perguntas e a tradução automática desempenham um papel crucial no avanço da capacidade máquina de interpretar e responder eficazmente à linguagem humana (Otter et al., 2019).

Recuperação e Extração de Informação

Sistemas de recuperação de informação (*Information Retrieval* (IR)) têm como objetivo ajudar os utilizadores a encontrar a informação mais relevante no formato mais conveniente e no momento certo. Um desafio importante em IR é a classificação de documentos com base na relevância para uma consulta. Os modelos de aprendizagem profunda (DL) para *ad hoc retrieval* centram-se na produção de representações das interações entre palavras individuais na consulta e nos documentos, seja através de abordagens focadas na representação ou focadas na interação. Avanços como o modelo *Deep Relevance Matching Model* (DRMM) (Guo et al., 2016) e o sistema *Standalone Neural Ranking Model with Pseudo-Relevance Feedback* (SNRM PRF) (Zamani et al., 2018) melhoraram a classificação de documentos, alcançando resultados de última geração (Otter et al., 2019). A extração de informação (*Information Extraction*) é a área de NLP que se dedica a extrair informação explícita ou implícita do texto. Os dados e as relações extraídas são frequentemente armazenados em bases de dados relacionais. As tarefas comuns incluem a extração de eventos, a extração de relações e o reconhecimento de entidades nomeadas (NER) (Otter et al., 2019).

A extração de eventos (*Event Extraction*) centra-se na identificação de frases que se referem a eventos e na extração dos seus participantes, como agentes e objetos. Técnicas como CNNs e modelos baseados em RNN têm sido utilizadas para detetar gatilhos de eventos (*event triggers*) e argumentos com sucesso significativo em vários testes de performance (*benchmarks*) (Otter et al., 2019).

A tarefa de extração de relações (*relationship extraction*) identifica relações como posse ou sinonímia entre elementos de uma frase. Abordagens iniciais para esta tarefa de NLP utilizaram CNN. No entanto, modelos mais avançados como o *BiLSTM-CNN* (Shan et al., 2021) e *Gated Recurrent Unit*s (GRUs) baseados em mecanismos de atenção melhoraram significativamente a extração de relações no texto (Otter et al., 2019).

O reconhecimento de entidades nomeadas (NER) identifica os nomes próprios e informações como datas, horas e preços no texto. Vários modelos, como o *Long Short Term Memory network* (LSTM), *CharWNN* (dos Santos and Guimarães, 2015) e *BiLSTM-CRF* (Ma and Hovy, 2016), alcançaram um desempenho de última geração em NER em diversos idiomas (Otter et al., 2019).

Classificação Textual

A classificação textual (*text classification*) atribui texto a categorias predefinidas, existindo múltiplas áreas de aplicação. Modelos baseados em CNN e arquiteturas híbridas, utilizando redes de crenças profundas (*Deep Belief Network*s (DBNs)) mostraram melhorias nas tarefas de classificação ao nível da frase e do documento. Abordagens recentes com a utilização do modelo *BERT* avançaram ainda mais o estado da arte na classificação de texto (Otter et al., 2019).

Geração de Texto

A geração de texto (*text generation*) envolve a criação de resultados em linguagem semelhantes à linguagem humana, onde se inclui tarefas como resumo automático, sistemas de resposta a perguntas e a tradução automática. Os modelos codificadores-descodificadores com mecanismos de atenção têm sido amplamente bem-sucedidos na geração de texto coerente e contextualmente preciso (Otter et al., 2019).

O resumo automático (*summarization*) reduz um documento de texto ao seu conteúdo mais importante, existindo dois tipos de resumo: extrativo, que seleciona frases do texto original e a abstrativo, que gera resumos a partir do resumo do conteúdo original. A utilização de modelos neuronais que recorrem a mecanismos de atenção melhoraram o desempenho da resumo abstrativo (Otter et al., 2019).

Os sistemas de resposta a perguntas (*question answering*) recuperam texto relevante de um documento para responder a uma consulta. Os modelos recentes utilizam mecanismos de atenção e redes de ponteiros (*pointer networks*) para combinar as perguntas com as passagens que contêm as respostas e refinar a representação da máquina para previsões precisas (Otter et al., 2019).

A tradução automática (*machine translation*) converte texto de um idioma para outro distinto. Os modelos de tradução automática neuronal (*Neural Machine Translation* (NMT)) utilizam arquiteturas codificador-descodificador com mecanismos de atenção para lidar com as complexidades da tradução de texto entre línguas (Otter et al., 2019).

Atribuição de Entidades

A tarefa de atribuição de entidades (*Entity Linking* (EL)) consiste em associar menções de entidades em textos não estruturados às suas entidades correspondentes numa base de conhecimento (*Knowledge Base* (KB)), sendo a *Wikipedia* a mais utilizada para este fim (Zhang et al., 2024a; Zhang, 2024). Os modelos mais recentes abordam este desafio através da criação de representações de entidades a partir de um subconjunto de informações da base de conhecimento, como descrições de entidades, tipos de entidades de alta granularidade, ou geração auto regressiva do texto de entrada. Embora a tarefa de EL esteja bem

estabelecida no domínio da *Wikipedia* no contexto de NLP, esta permanece sub explorada para outras áreas, como do mercado de trabalho (Zhang et al., 2024a; Zhang, 2024). Um dos primeiros modelos generativos usados para atribuição de entidades foi o *GENRE*, que propôs uma metodologia de sequência-para-sequência que gera uma uma sequência de rótulos de entidades a partir de uma sequência de menções condicionada com alguns indicadores especiais. No entanto, como a maioria dos trabalhos existentes de atribuição de entidades, o modelo *GENRE* requer treino completo do zero, o que exige uma quantidade enorme de dados e recursos de hardware (Ding et al., 2024). Os métodos supervisionados falham quando os dados são ruidosos, incompletos, mal descritos ou raros, tendo menos exemplos para as entidades menos frequentes. Trabalhos recentes encontraram que pelo menos 5% dos rótulos de verdade (*ground truth*) no conjunto de dados *CONLL03* (Sang and Meulder, 2003) estão incorretos. Da mesma forma, pelo menos 10% dos rótulos de verdade na tarefa de atribuição de entidades do *CONLL03* (Sang and Meulder, 2003) estão provavelmente incorretos (Ding et al., 2024).

2.2 Estado da Arte na Seleção de Recursos Humanos

A crescente complexidade no processo de seleção de candidatos, exacerbada pela alta competitividade e pela diversificação das habilidades exigidas no mercado de trabalho, impulsiona o uso de tecnologias inovadoras como a AI, em especial os LLMs, para otimizar a avaliação de competências e qualificações. Nesse cenário, a adoção de LLMs no processamento de linguagem natural apresenta-se como uma solução promissora para resolver os desafios enfrentados por abordagens tradicionais de automação no recrutamento. Este capítulo visa fornecer uma análise abrangente dessa dinâmica evolutiva, explorando como a adoção de tecnologias como os LLM se compara e permite resolver problemas associados a abordagens tradicionais de automação no processo seletivo. É imperativo analisar criticamente o estado atual da arte relativo à forma como os LLM têm sido utilizados para auxílio e como meio de ferramenta para processos de seleção de candidatos.

A adequação pessoa-trabalho é uma componente essencial das plataformas de recrutamento, atendendo a diversas aplicações subsequentes, como a procura de emprego e a recomendação de candidatos que melhor se adequem a uma determinada vaga (Cao et al., 2024). Um sistema competente na adequação pessoa-trabalho pode reduzir significativamente as taxas de desemprego, diminuir os custos elevados de recrutamento e eliminar os esforços despendidos nos processo de recrutamento (Cao et al., 2024).

2.2.1 Sistemas Tradicionais de Automatização da Seleção de Recursos Humanos

Abordagens tradicionais de automação da triagem e seleção de candidatos, implementadas em sistemas de *Applicant Tracking System* (ATS), são globalmente usadas por recrutadores e recursos humanos, sendo que 75% dos recrutadores usa sistemas de ATS e 98% das empresas *Fortune 500*, que contém as 500 maiores corporações dos Estados Unidos, aplicam algum sistema ATS nos seus processos de recrutamento (Koh et al., 2023). Estes sistemas tem como principal função ajudar os recursos humanos a processar grandes quantidades de candidaturas ao filtrar os candidatos que não correspondem aos critérios definidos (Koh et al., 2023).

Estes sistemas baseiam-se na extração dos dados dos currículos dos candidatos e de metadados das descrições de emprego para efetuar a comparação da informação por palavras-chave (*keyworkds*) e critérios definidos (*features*). Estes são muitas vezes usados em metodologias como filtragem colaborativa (*collaborative filtering*) para sistemas de recomendação de empregos (Koh et al., 2023; Skondras et al., 2023; Cao et al., 2024). Estes sistemas tratam a correspondência (*match*) entre o CV e a descrição da vaga como um problema de correspondência de texto supervisionada (*supervised text-matching*), com a utilização de dados de treino (Wu et al., 2023).

Os sistemas tradicionais ATS auxiliaram no desenvolvimento de abordagens baseadas em modelos de DL para gerar *embeddings* textuais a partir de dados em grande escala dos currículos dos candidatos e das descrições das vagas de emprego (Cao et al., 2024). Estes modelos podem também ser baseados em PLMs, que dependem de parâmetros textuais de entrada e retornam etiquetas (*labels*) para ser usados tarefas de classificação de currículos (Gan et al., 2024).

Porém, sistemas tradicionais de ATS apresentam diversas limitações e desafios, entre os quais a potencial inerente discriminação e viés (*bias*) de candidatos, uma vez que estes sistemas dependem da procura por palavras-chave e critérios definidos pelos recursos humanos, que podem incorporar involuntariamente preconceitos inconscientes nos algoritmos (Koh et al., 2023). Este viés pode resultar na exclusão de candidatos qualificados, particularmente de grupos sub-representados ou com percurso de carreira não linear ou padronizada (Koh et al., 2023; Skondras et al., 2023). Para além disso os algoritmos tradicionais de ATS tendem a favorecer candidatos de escolas prestigiadas com maiores medias de notas, o que nem sempre se correlaciona como um indicador para o sucesso numa determinada vaga (Koh et al., 2023). O que realça a necessidade de processos de avaliação meticulosos para garantir justiça e imparcialidade na classificação de currículos.

Outro grande desafio na utilização de sistemas tradicionais de ATS é a lacuna semântica existente entre os CVs e as descrições das vagas de emprego (Zheng et al., 2023). Além disso, a grande variação no formato e na estrutura dos CVs, que podem ser não es-

truturados ou semi-estruturados, contribui para a imprecisão na extração e compreensão das informações presentes tanto nos CVs quanto nas descrições das vagas. Essa imprecisão muitas vezes exige a intervenção manual no processo de triagem (Skondras et al., 2023; Gan et al., 2024), que seria indesejável por poder introduzir erros ou viés. Métodos tradicionais têm dificuldades em lidar com a variabilidade na apresentação dos currículos, por estes não seguirem um formato padrão (Skondras et al., 2023), exigindo técnicas mais aprimoradas de NLP e mecanismos de classificação mais sofisticados para melhorar a precisão e a eficiência dos processos de recrutamento (Gan et al., 2024).

A escassez e a baixa qualidade dos dados dos CVs e das descrições das vagas de emprego, em muitas categorias do mercado de trabalho, afeta o desempenho dos algoritmos de ATS e dos sistemas de recomendação de vagas de emprego (Skondras et al., 2023). Além disso, a escassez de dados de alta qualidade, bem anotados, gera dificuldades significativas na compreensão e extração de habilidades e competências dos CVs e das descrições de vagas de emprego. Estas limitações também impactam negativamente o treino de modelos robustos, pois a insuficiente quantidade de dados representativos compromete a capacidade destes modelos em generalizar de forma eficaz. Como consequência, a correspondência entre CVs e descrições de vagas e a tarefa de classificação de currículos apresenta um desempenho sub ótimo nos métodos tradicionais de ATS (Skondras et al., 2023; Du et al., 2023). A qualidade e o tamanho dos dados de treino são também fatores cruciais para a eficácia dos modelos de aprendizagem automática, como o *BERT* e redes neuronais convulsionais (CNNs). No entanto, a obtenção de grandes conjuntos de dados anotados continua a ser um desafio significativo, limitando assim a eficácia desses modelos. Esta limitação sugere que, sem dados de alta qualidade e em quantidade adequada, os modelos poderão não alcançar o desempenho desejado, especialmente em contextos de aplicação prática (Skondras et al., 2023).

As abordagens tradicionais de correspondência entre currículos e vagas de emprego nos sistemas de ATS geralmente tratam essa tarefa como um problema de correspondência de texto supervisionada (*supervised text-matching*), que depende diretamente de pares de dados para treinar os modelos de recomendação. No entanto, esses métodos enfrentam dificuldades relacionadas à escassez de dados, o que compromete a eficácia das abordagens tradicionais, evidenciando a necessidade de melhorias nos sistemas de recomendação de empregos para lidar de forma mais eficaz com esses desafios (Wu et al., 2023).

Adicionalmente, os processos de extração de habilidades em sistemas tradicionais de ATS muitas vezes falham em normalizar sinónimos e habilidades parafraseadas, resultando em análises inconsistentes do mercado de trabalho. Tratar a extração de habilidades como uma tarefa de classificação de múltiplos rótulos extremos (*Extreme Multilabel Classification* (XMLC)) torna o desenvolvimento de conjuntos de dados de treino de alta qualidade ainda mais desafiador, afetando a robustez desses modelos (Decorte et al., 2023).

Por fim, os sistemas de ATS tradicionais são frequentemente criticados pela dependência de técnicas de correspondência de palavras-chave, que geralmente não conseguem capturar as informações com nuances e contexto presentes nos dados (Skondras et al., 2023).

Estas limitações sugerem a necessidade de sistemas de recrutamento mais avançados, transparentes e justos que possam avaliar os candidatos de forma precisa e holística, ao mesmo tempo em que enfrentam as deficiências atuais nos ATS tradicionais.

2.2.2 Integração de LLMs na Automatização da Seleção de Recursos Humanos

A AI tem sido amplamente utilizada para mitigar as limitações dos sistemas tradicionais de ATS, e auxiliar a automação da triagem e seleção de currículos, sendo um aspeto crucial do processo de recrutamento nas organizações. Os sistemas automatizados de triagem de currículos são baseados em diversas tarefas de NLP (Gan et al., 2024; Phillips and Robie, 2024). Esses modelos de NLP são essenciais para a aplicação e automação de tarefas de classificação de currículos e extração de conteúdo de CVs ou descrições das vagas (Gan et al., 2024; Phillips and Robie, 2024).

Embora existam muitos estudos que exploram o uso de AI em processos de seleção, o conhecimento sobre como os LLMs podem ser empregues para aumentar a fiabilidade desses sistemas ATS ainda é limitado (Phillips and Robie, 2024). Reconhece-se no entanto a promissora potencialidade dos LLM em melhorar os sistemas tradicionais de ATS e de recomendação de candidatos a vagas de emprego (Gan et al., 2024; Du et al., 2023; Wu et al., 2023), uma vez que estes demonstram grande capacidade de criação semântica e amplo conhecimento externo (Wu et al., 2023), assim como uma robusta capacidade de generalização em diversas tarefas NLP (Gan et al., 2024).

Os LLMs têm demonstrado melhorias significativas nas tarefas de NLP, evidenciando sólidas capacidades de extração de informação; anotação de dados; classificação de dados e geração de conteúdo. Devido a estas capacidades, os LLMs podem ser utilizados para gerar dados de treino ou *datasets* para aprimorar outros modelos de AI (Gan et al., 2024; Zheng et al., 2023; Koh et al., 2023; Du et al., 2023). Estes avanços sugerem um potencial promissor para a aplicação de LLMs em sistemas ATS e de recomendação, visando não apenas otimizar a eficiência, mas também aumentar a precisão e a equidade nos processos de seleção (Gan et al., 2024; Zheng et al., 2023; Koh et al., 2023; Du et al., 2023). LLMs avançados, como *GPT-3.5*, *GPT-4* e *LLama2*, rapidamente suplantaram esforços individuais de modelagem que permeavam a comunidade de NLP há anos (Ding et al., 2024), uma vez que o desempenho destes modelos em configurações *zero-shot* e *few-shot* rapidamente substituíram abordagens de pré-treino e *finetuning* para poder resolver com maior eficiência tarefas de NLP. Recentemente, têm sido feitos esforços no desenvolvimento

de soluções mais eficientes para a criação de *prompts* que permitam aos LLMs executar tarefas de NER e EL de forma mais eficaz (Ding et al., 2024).

A capacidade dos LLM em extrair informações tem sido empregue na análise de currículos, visando identificar secções específicas, tais como a experiência profissional do candidato, formação académica, habilidades, competências, além de características comportamentais presentes nos perfis dos candidatos e dos metadados dos empregos (Skondras et al., 2023; Cao et al., 2024). O trabalho de Cao et al. propôs o método *TAROT* para melhorar a utilização de informação estrutural e semântica em LLMs, tendo como alvo específico o texto semi-estruturado encontrado em perfis e descrições de empregos, utilizando tarefas de pré-treino multi-granulares para capturar e refinar a informação semântica. O modelo opera de forma hierárquica, começando com *embeddings* ao nível de frase extraídos pelo modelo *BERT* e progredindo através de níveis cada vez mais complexos, culminando numa camada de atenção cruzada (*cross-attention*) que melhora a interação entre os *embeddings* da descrição da vaga de emprego e do perfil do candidato. Esta abordagem hierárquica permite que o método *TAROT* adapte os *embeddings* com base nas necessidades, melhorando assim a precisão e a relevância das avaliações de adequação pessoa-emprego. As experiências realizadas em conjunto de dados reais demonstraram melhorias significativas no desempenho, destacando a sua eficácia em tarefas de adequação pessoa-emprego e a superioridade em relação a outros métodos tradicionais de ATS (Cao et al., 2024).

Foi desenvolvido por Skondras et al., uma abordagem inovadora que se apoia na utilização de LLMs em sistemas de recomendação de empregos, denominada de *Generative job Recommendation paradigm based on LLM* (*GIRL*) (Skondras et al., 2023). A metodologia utilizada diverge de modelos tradicionais ao focar-se na geração de descrições de emprego personalizadas adaptadas aos currículos dos candidatos. O sistema baseia-se na utilização de um LLM concebido através de um processo de treino que consistiu em três etapas; Na etapa de *Supervised Fine-Tuning*, onde envolveu o treino do LLM para gerar descrições de emprego adequadas com base nos CVs fornecidos, onde foi utilizado um *dataset* que continha pares de CVs e descrições de vagas de emprego; Na etapa de treino do Modelo de Recompensa, onde este foi treinado utilizando um *dataset* de dados contendo tanto pares CVs e descrições de vagas de emprego correspondentes como não correspondentes, incluindo a opinião (*feedback*) dos recrutadores. Este modelo foi desenvolvido para avaliar e pontuar a relevância das descrições de emprego geradas, imitando o processo de tomada de decisão de um recrutador; Na ultima etapa de aprendizagem por reforço com *feedback* de recrutadores, necessário para refinar ainda mais o LLM, foi empregue a aprendizagem por reforço baseada em *Proximal Policy Optimization*(*PPO*). Esta etapa alinha texto de saída (*output*) do LLM com as preferências dos recrutadores, considerando tanto as necessidades dos candidatos como as exigências do mercado (Skondras et al., 2023). O sistema proposto foi testado em conjuntos de dados reais, demonstrando

melhorias significativas nos sistemas de recomendação, na precisão das correspondências e na eficácia geral das recomendações de emprego (Skondras et al., 2023). No entanto, treinar LLMs para tarefas de recomendação é uma tarefa não trivial, onde dadas as diferenças significativas entre tarefas de recomendação e tarefas de NLP, o LLM precisa de incorporar grande conhecimento de domínio específico (Zheng et al., 2023). Tendo os LLM um design orientado para um domínio geral, sendo estes treinados em grandes *corpus* genéricos, enfrentam dificuldades em capturar a informação estrutural única contida no contexto do mercado de trabalho, podendo resultar em perdas de relações estruturais e semânticas latentes nos perfis dos candidatos e nas descrições das vagas de emprego (Cao et al., 2024). Este desalinhamento leva à perda de correlações semânticas latentes importantes, essenciais para tarefas como o ajuste dos candidatos certos às vagas de emprego. Além disso, a estrutura hierárquica das descrições de emprego e dos currículos que é concebida para transmitir informações específicas de diversos domínios não é devidamente processada por estes modelos. Isso resulta no colapso das representações geradas pelos LLM quando aplicadas a este tipo de dados semi-estruturados, especialmente quando não é utilizado pré-treino específico para o domínio (Cao et al., 2024).

É importante também ter em consideração que utilizar LLMs em processos de recomendação ou para simular a tomada de decisão de um recrutador, apesar de oferecer um potencial significativo para automatizar o processo de triagem de candidaturas, a sua dependência de dados de treino derivados de texto real introduz a potencial perpetuação de preconceitos inerentes, como desinformação e manipulação em sistemas automatizados. Estes preconceitos, que refletem os sentimentos e crenças da sociedade, apresentam riscos que podem afetar a equidade dos processos de recrutamento (Koh et al., 2023; Phillips and Robie, 2024). Além disso, os LLM têm maior dificuldade em simular a característica de conscienciosidade, que pode ser atribuído à ênfase na extroversão, vista como um traço mais estereotípico na descrição do trabalho, enquanto que conectar a conscienciosidade à descrição fornecida seria mais subtil (Phillips and Robie, 2024). Embora seja necessário lidar com os preconceitos presentes nestes modelos, pode não ser totalmente possível eliminá-los (Koh et al., 2023), o que apresenta um desafio crítico no uso dos LLMs no recrutamento, onde a equidade é fundamental (Koh et al., 2023; Phillips and Robie, 2024). É necessário mais desenvolvimento cientifico de pesquisa, especialmente na redução de vieses (*biases*), preconceitos e do poder discriminatório dos modelos, para ser possível garantir que os LLM não geram resultados injustos ou discriminatórios, principalmente quando utilizados como mecanismos de tomada de decisão (Koh et al., 2023).

No trabalho realizado por Gan et al. foram utilizados agentes com recurso a LLMs, que desempenhavam funções de NLP para a extração de informação (NER), resumo e classificação de currículos. O LLM foi ajustado (*fine-tuned*) para melhorar o desempenho nessas tarefas, aumentando a precisão e eficiência do processo de triagem de currículos. Além disso, os agentes demonstraram grande potencial na automatização de tarefas essen-

ciais em sistemas de ATS, como a categorização de frases e a identificação de competências e experiências profissionais relevantes, sublinhando a sua utilidade na melhoria dos processos de recrutamento (Gan et al., 2024). No entanto, o autor alerta que a utilização de LLMs de código fechado, desenvolvidos por grandes corporações, levanta preocupações significativas no que diz respeito à segurança e privacidade (Gan et al., 2024). O principal problema reside na potencial fuga de informações privadas, quer relacionadas com os dados utilizados para treinar esses modelos, quer com os dados fornecidos pelos candidatos como entrada (*input*), o que aumenta o risco de comprometimento dessas informações. Esta questão torna-se especialmente crítica em aplicações sensíveis, como a triagem de currículos onde estão envolvidas informações pessoais confidenciais.

Trabalhos de diversos autores Zheng et al.; Du et al.; Decorte et al. visaram explorar o uso de LLMs para a geração de *datasets* de treino contendo currículos sintéticos, através do recurso a técnicas de engenharia de *prompts*. Estes dados foram usados para treinar modelos de NLP presentes nos sistemas de ATS para melhorar a performance em tarefas de classificação de currículos (Zheng et al., 2023) e para enfrentar o desafio da extração de competências em anúncios de emprego, tarefa que representa um problema de classificação extrema com múltiplas etiquetas (*Extreme Multilabel Classification* (XMLC)), onde não existe um grande conjunto de dados devidamente anotado (Decorte et al., 2023). No trabalho realizado por Decorte et al. foi usado um LLM para gerar frases associadas a competências específicas da taxonomia ESCO, criando um *dataset* com pares de treino "competência" e "frase" necessários para o treino do modelo. Em vez de extrair competências do texto, o modelo foi incitado a gerar frases de anúncios de emprego com base em competências predefinidas, contornando assim a dificuldade de alinhar a saída com a ontologia das competências. O treino do modelo foi efetuado com a abordagem de aprendizagem contrastiva (*contrastive learning*), onde foi empregue a arquitetura *bi-encoder* para codificar tanto as competências como as frases relacionadas com o emprego. Estes pares codificados foram otimizados num espaço vetorial partilhado com o modelo para aprender a aproximar os nomes das competências e das frases correspondentes com base na similaridade por cosseno. Os resultados da aplicação da metodologia demonstram o potencial dos LLMs na melhoria da precisão da extração de competências em sistemas de recrutamento. Ao sintetizar dados de treino e aproveitar a aprendizagem contrastiva (*contrastive learning*) esta abordagem melhorou a capacidade do LLM em mapear com precisão os requisitos da vaga de emprego para as competências da taxonomia ESCO relevantes, sem a necessidade de grandes conjuntos de dados anotados, tornando-se uma solução altamente eficiente para tarefas de recrutamento (Decorte et al., 2023). No entanto, a utilização de LLMs para a criação de dados sintéticos de treino apresenta várias limitações, nomeadamente em relação à geração de conteúdo fabricado e aos problemas de *few-shot*, que impactam negativamente a qualidade do conteúdo gerado (Du et al., 2023). Os LLMs frequentemente produzem informações fabricadas ou incompletas, de-

vido a falhas na compreensão dos perfis dos candidatos e das descrições de emprego, especialmente quando os dados de interação disponíveis são insuficientes. Este problema é agravado pelo efeito de cauda longa (*long-tail*), em que candidatos com registos de interação limitados fornecem pouca informação, levando os LLMs a gerar recomendações imprecisas ou alucinatórias (ou seja, conteúdo errado ou sem base lógica no contexto do texto analisado) (Du et al., 2023). Por outro lado, candidatos com dados esparsos tendem a sofrer com a geração de currículos de qualidade inferior. Para mitigar estes problemas têm sido exploradas soluções como o uso de redes adversarias gerativas (*Generative adversarial network*s (GANs)), uma arquitetura de DL que treina dois modelos, um gerador que cria novos exemplos plausíveis, e um discriminador que diferencia entre dados reais e fabricados, permitindo no final a geração de dados realistas. Esta abordagem permitem alinhar currículos de baixa qualidade com currículos de alta qualidade, melhorando assim a precisão das recomendações. No entanto, apesar dessas melhorias, os LLMs continuam a enfrentar dificuldades relacionadas com lacunas de conhecimento, exigindo medidas adicionais, tais como a engenharia de *prompts* e aprendizagem em contexto (*in-context learning*), de forma a minimizar alucinações e a adaptar-se de forma mais eficaz a tarefas de recomendação de domínios específicos, sem necessidade de ajustes extensivos nos parâmetros do modelo (Du et al., 2023).

Segundo o autor Ding et al. que explorou a capacidade dos LLMs em tarefas de NLP, a utilização destes modelos em tarefas de extração de informação (*Information Extraction*), reconhecimento de entidades nomeadas (NER) e EL demonstram limitações significativas quando o número de entidades ultrapassa algumas dezenas. Embora os LLMs apresentem um desempenho promissor em contextos com um número limitado de entidades, a sua eficácia diminui drasticamente à medida que a complexidade e a quantidade de entidades aumentam (Ding et al., 2024).

No trabalho de Zhang et al., foram propostos dois novos modelos de NLP para EL baseados nos modelos *GENRE* (Cao et al., 2021) e *BLINK* (Li et al., 2020) para a base de conhecimento (*Knowledge Base* (KB)) da ESCO, que atribuí uma *Skill* na taxonomia ESCO a uma menção de uma entidade *Skill* extraída do texto. Este processo é crucial para quantificar as dinâmicas do mercado de trabalho e entender as necessidades e demandas de *Skills* específicas (Zhang et al., 2024a; Zhang, 2024). A extração de *Skills* e o vínculo com uma entrada única da taxonomia ESCO permite uma visão mais precisa do mercado. Segundo a avaliação do autor, este processo deve ser aprimorado, pois a eficácia atual não é satisfatória, particularmente no caso das chamadas "*Skills* implícitas", onde as menções no texto não correspondem exatamente a uma *Skill* da taxonomia, tornando a correspondência menos precisa. Assim, Zhang sugere a investigação no desenvolvimento do estado da arte dos LLMs no campo de EL (Zhang, 2024). O que motiva a investigação mais aprofundada no desenvolvimento de modelos mais robustos e eficazes para a tarefa de EL, especialmente com o uso de LLMs e abordagens modernas que lidam tanto com

Skills explícitas quanto implícitas. A utilização de heurísticas baseadas em distância de *Levenshtein* e similaridade semântica entre frases também é destacada como uma abordagem promissora para lidar com casos onde não há uma correspondência exata (Zhang et al., 2024a). Essa linha de investigação sugere um caminho promissor para o desenvolvimento do estado da arte no campo de EL com a utilização de LLMs, possibilitando uma atribuição mais precisa e eficaz de entidades *Skills* no contexto de mercado de trabalho. O trabalho de Zhang et al. apresenta outras limitações importantes que impactam a avaliação de desempenho dos modelos de EL. A principal limitação está no processo de avaliação, que considera apenas um rótulo correto por entidade mencionada, o que pode levar a uma subestimação do desempenho dos modelos, uma vez que várias previsões "tecnicamente corretas" acabam por ser penalizadas por não corresponderem exatamente a um rótulo único. Além disso, o uso de dados sintéticos para treino pode não capturar completamente a diversidade e as nuances presentes em documentos reais de anúncios de emprego (Zhang et al., 2024a).

Futuras direções para mitigar os problemas inerentes aos LLMs no contexto do recrutamento centram-se, em grande parte, na necessidade de lidar com preconceitos e limitações técnicas, com vista a melhorar a equidade e a eficácia das sistemas de ATS, sendo uma necessidade aumentar a consciencialização sobre os preconceitos inerentes a estes modelos e incentivar o desenvolvimento de sistemas de AI mais transparentes e responsáveis, com o objetivo de mitigar os seus impactos negativos. É fundamental manter a investigação contínua em métodos que possam reduzir eficazmente os preconceitos e garantir que as ferramentas de recrutamento baseadas em LLMs promovam processos justos e equitativos (Koh et al., 2023).

Uma abordagem promissora para melhorar a precisão dos LLMs em tarefas de classificação de múltiplas classes de currículos passa pela adição de camadas de classificação, onde estas capturarem de forma mais eficaz características específicas e corrijam eventuais erros de classificação (Skondras et al., 2023). A fusão de dados reais com dados sintéticos gerados por LLMs como o *ChatGPT* pode ser uma solução interessante em tarefas como extração de metadados e reconhecimento de entidades (NER), em que seja necessário conhecimento especifico de um domínio fazendo impulsionar significativamente a eficácia dos sistemas (Skondras et al., 2023). Além disso, os LLMs podem ser aproveitados para identificar padrões complexos em dados não estruturados, como descrições de empresas e anúncios de emprego, otimizando assim o processo de correspondência entre currículos e vagas (Skondras et al., 2023). Estes avanços têm potencial de melhorar substancialmente os sistemas de ATS, sendo aplicáveis em outros domínios.

Outro desafio significativo na aplicação de LLMs no recrutamento reside na extração de competências. Os LLMs apresentam dificuldades em refletir múltiplas competências numa única representação, dado que as frases sintéticas tendem a focar-se numa competência única, em contraste com frases do mundo real que frequentemente mencionam

várias (Decorte et al., 2023). Esta limitação pode prejudicar o desempenho dos modelos de extração de competências, comprometendo a precisão das recomendações. Adicionalmente, é crucial monitorizar e mitigar os preconceitos que possam surgir durante a geração de dados, de forma a evitar resultados injustos ou discriminatórios. Para garantir que as futuras melhorias estejam alinhadas com princípios de equidade, é importante definir e avaliar a equidade no contexto específico das competências extraídas e da sua aplicação (Decorte et al., 2023).

Compreender as limitações e potencialidades dos LLM é crucial para o desenvolvimento de melhores práticas e metodologias na utilização destas tecnologias em contextos do mercado de trabalho e seleção de recursos humanos.

Metodologia

Formulação da Questão de Pesquisa

No desenvolvimento da questão de pesquisa da revisão bibliográfica, foi aplicada a metodologia *Patient or Population, type of Interventions, Comparisons and Outcomes* (PICO), amplamente reconhecida por sua eficácia na formulação de perguntas de pesquisa científicas relevantes e bem definidas. A estrutura PICO é composta por quatro elementos principais: o Problema, Processo ou Paciente (P), que no contexto de engenharia, geralmente se refere a um problema técnico ou de processo em vez de uma população; a Intervenção ou Melhoria (I), que se refere à solução ou intervenção proposta; as Comparações (C), caso existam, que envolvem a comparação com práticas atuais ou visões alternativas; e os Resultados (O) de interesse, que são os objetivos ou impactos esperados da intervenção. Esta abordagem é fundamental para estruturar a pergunta de pesquisa que a revisão se propõe a responder (Bettany-Saltikov, 2010).

A questão de pesquisa da revisão bibliográfica foi formulada utilizando a metodologia PICO, cujos elementos são definidos da seguinte maneira:

- **Problema**: A complexidade na seleção de candidatos por parte das empresas.
- **Intervenção**: Utilização de ferramentas de *Artificial Inteligence* nomeadamente de processamento de linguagem natural, especialmente *Large Language Models*, para melhorar a análise e interpretação do contexto das informações apresentadas por cada recurso humano e das especificações e normas definidas por processos de Recursos Humanos (por exemplo, CVs).
- **Comparação**: Comparação entre as abordagens tradicionais de análise por correspondência de palavras-chave e outras estratégias de seleção automática com a abordagem proposta com o uso de AI e LLM.

- **Outcome/Resultado**: Avaliação das competências e qualificações dos recursos humanos, assim como a correspondência com requisitos específicos de seleção.

A partir da formulação PICO foi possível desenvolver a questão de pesquisa e da consequente questão booleana, a ser usada para pesquisa nas bases de dados de artigos científicos.

A **questão de pesquisa** resultante da aplicação da metodologia de PICO foi a seguinte: *Em empresas que enfrentam a complexidade na seleção de candidatos, como a utilização de Inteligência Artificial (AI), nomeadamente Large Language Models (LLM) no processamento de linguagem natural em comparação com a abordagens tradicionais de automação no processo de seleção afeta a avaliação das competências e qualificações dos recursos humanos, bem como a correspondência com requisitos específicos de seleção?*

Expressão Booleana Resultante: A expressão booleana formulada através da aplicação da metodologia PICO e da formulação da questão de pesquisa, que será utilizada nas bases de dados científicas selecionadas, é a seguinte:

("large language models" OR "LLM") AND ("human resource selection" OR "recruitment" OR "candidate selection" OR "employee selection") AND ("recruitment" OR "selection" OR "hiring")"

A expressão booleana foi usada como pesquisa avançada bases de dados científicas, sobre as seguintes condições:

- *("large language models" OR "LLM")*, pesquisado em todos os metadados das fontes procuradas ("*in all metadata*").
- *AND ("human resource selection" OR "recruitment" OR "candidate selection" OR "employee selection")*, pesquisado em todo os metadados das fontes procuradas ("*in all metadata*").
- *AND ("recruitment" OR "selection" OR "hiring")*, pesquisado no resumo das fontes procuradas ("*in abstract*").

Intervalo de Tempo de Pesquisa

Tendo em conta o desenvolvimento recente dos LLM foi escolhido o intervalo de tempo de pesquisa compreendido desde as primeiras aplicações significativas dos LLM, 2018, até ao momento aquando do desenvolvimento da pesquisa, Abril de 2024.

Bases de dados utilizadas

Como bases de dados foram escolhidas as seguintes:

- IEEE Xplore

- ScienceDirect
- b-on
- arXiv

Resultados da Pesquisa nas Bases de Dados

No desenvolvimento da revisão bibliográfica foram realizadas pesquisas nas bases de dados selecionadas com a questão (*query*) booleana definida pela a aplicação da metodologia PICO. Assim foi possível identificar os artigos relevantes para o estudo. Abaixo, estão descritos os resultados obtidos de cada base de dados:

Tabela 2.1: Resultados da Pesquisa nas Bases de Dados Científicas (2018-2024)

Base de Dados	Elementos Encontrados	Elementos Exportados com Sucesso
IEEE Xplore	2	2
arXiv	33	33
b-on	210	193
ScienceDirect	37	37

Os elementos encontrados referem-se aos artigos identificados no intervalo de 2018 a 2024, enquanto os elementos exportados com sucesso são aqueles que foram efetivamente incluídos na análise após o processo de exportação e revisão dos dados. Durante o processo de exportação dos artigos encontrados na base de dados *b-on*, alguns elementos foram removidos devido a problemas de exportação, resultando em 193 artigos válidos exportados com sucesso. No total, após a remoção de 36 artigos duplicados, a análise foi conduzida com 229 artigos únicos.

Tabela 2.2: Remoção de Duplicados e Total Final

Descrição	Quantidade
Duplicados Removidos	36
Total Final	229

Metodologia de Comparação e Seleção de Artigos

Foi utilizada a metodologia *Preferred Reporting Items for Systematic reviews and Meta-Analyses* (PRISMA) *Statement* para comparação e seleção (*screening*) dos artigos a serem incluídos na pesquisa, permitindo organizar e estruturar a estratégia de seleção. A formulação PRISMA fornece orientações claras para revisores sistemáticos sobre como relatar de forma transparente o porquê da revisão ter sido realizada, o que os autores fizeram e o que encontraram, os avanços feitos nos métodos identificados e como selecionar, avaliar e sintetizar estudos (Page et al., 2021).

A utilização da metodologia PRISMA permitiu delinear de forma estruturada os processos de identificação, triagem e seleção de artigos válidos, bem como a decisão final sobre os estudos que seriam incluídos na revisão. A utilização do PRISMA garantiu a abordagem sistemática e transparente na comparação e escolha dos artigos relevantes para o desenvolvimento deste trabalho. Na primeira fase da aplicação da metodologia PRISMA foram identificados e anotados os registos provenientes das bases de dados selecionadas. Em seguida, foram removidos os registos que não passam para a fase de seleção, pela aplicação de critérios como a remoção de duplicados e de registos considerados ilegíveis. Após a filtragem inicial, passou-se à segunda fase de triagem (*screening*) na aplicação da metodologia, onde se realizaram sucessivos processos de análise baseados em critérios definidos, como a exclusão de registos após a leitura do título ou resumo, eliminação de artigos não legíveis e remoção de registos que não atendiam às palavras-chave estabelecidas. No final desse processo, restaram os registos considerados elegíveis e acessíveis. Na última fase de inclusão todos os registos restantes foram revistos detalhadamente, sendo os mais relevantes selecionados para fazer parte da pesquisa.

A aplicação da metodologia PRISMA foi feita na plataforma Rayyan.ai, que permitiu adicionar etiquetas (*tags*) aos artigos para facilitar o processo de seleção e exclusão.

A seguir, são apresentados os resultados obtidos:

- Excluídos por palavras-chave: "clinical" e "medical": 25
- Excluídos por serem "estudo errado ou área errada" após leitura do *abstract*, mas encontrados pela consulta booleana: 179
- Restantes para leitura completa: 25

Relatório do Rayyan.ai:

- Total de artigos: 229
- Excluídos por recurso a ferramentas de filtro por palavras-chave "clinical" ou "medical": 25
- Excluídos após leitura do titulo e *abstract*: 179
- Excluídos após leitura completa: 8
- Total de excluídos: 212
- Artigos marcados com a *tag* "Talvez" de serem incluídos na pesquisa: 8
- Artigos marcados com a *tag* "Incluír" na pesquisa: 9

Com base na metodologia PRISMA, os artigos foram cuidadosamente filtrados e selecionados. Inicialmente foram encontrados 229 artigos. Desses, 25 foram excluídos por conterem palavras-chave relacionadas a "clinical" ou "medical". Outros 179 artigos foram excluídos por serem considerados estudos errados ou de áreas erradas, não se enquadrando na área de pesquisa, apesar de terem sido encontrados pela pergunta booleana. Após essas exclusões restaram 25 artigos para leitura completa.

Do total de artigos adicionados no Rayyan.ai, 8 foram excluídos após a leitura completa, resultando no total 212 artigos excluídos. Além disso, 8 artigos foram classificados como "talvez", e 9 artigos foram incluídos na revisão final.

Diagrama de fluxo PRISMA

O diagrama da Figura 2.1 mostra o processo detalhado de seleção e exclusão de artigos conduzido ao longo da revisão sistemática desta pesquisa, conforme os princípios da metodologia PRISMA. O fluxo descreve as várias etapas do processo, desde a identificação inicial dos artigos através de bases de dados até a triagem, avaliação de elegibilidade e inclusão final.

O processo começa com a identificação de artigos em cada base de dados selecionada, totalizando de 282 artigos encontrados. Em seguida, foi realizada uma exclusão prévia ao processo de triagem (*screening*), tendo sido excluídos 36 artigos duplicados e 17 artigos por não serem acessíveis, sendo considerados inelegíveis, restando 229 artigos para triagem. Desses 229 artigos, 179 foram excluídos após a leitura do título e resumo (*abstract*), resultando em 50 artigos para análise, sendo todos estes considerados acessíveis e portanto legíveis. Após a aplicação de critérios de exclusão, como a presença de palavras-chave irrelevantes à pesquisa foram excluídos 25 artigos como sendo desadequadas ao tema de estudo. Resultando na inclusão final de 25 artigos na revisão para leitura completa, onde 9 foram incluídos no estudo. A utilização do diagrama de fluxo PRISMA foi essencial para garantir a transparência e a reprodutibilidade do processo de seleção da revisão sistemática.

Ferramentas Utilizadas

Para o desenvolvimento do estado da arte foram utilizadas as seguintes ferramentas de apoio: o Mendeley para a organização dos artigos e das respetivas referências bibliográficas, permitindo uma gestão eficiente e estruturada das fontes; e o site Rayyan.ai usado para a filtragem e seleção dos artigos a serem incluídos ou excluídos da revisão bibliográfica, seguindo a metodologia PRISMA que facilita a aplicação sistemática dos critérios de inclusão e exclusão.

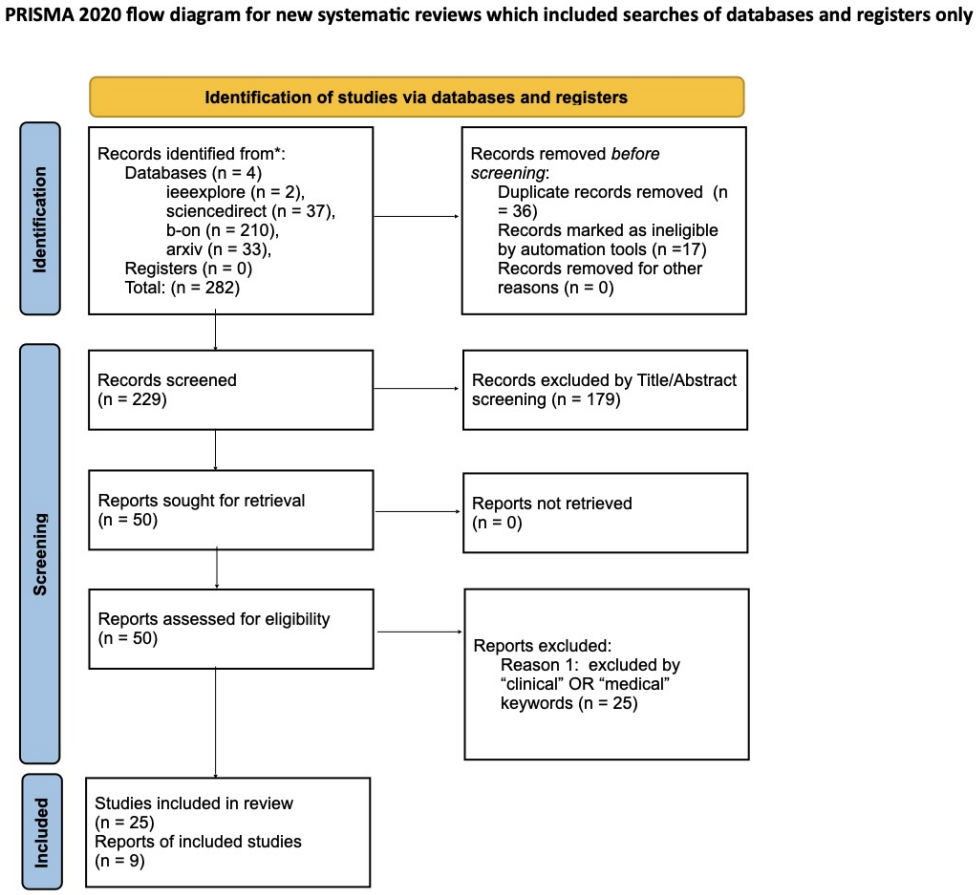


Figura 2.1: Diagrama de Fluxo PRISMA 2020 adaptado para o processo de seleção e exclusão de artigos na pesquisa

Capítulo 3

Sistema Assistido para Seleção de RH (SA4S-RH)

3.1 Requisitos e Arquitetura do sistema SA4S-RH

Neste capítulo é apresentado o sistema proposto no trabalho, assim como os requisitos do sistema, tecnologias utilizadas e a respetiva arquitetura. São descritos todos os passos de desenvolvimento e justificação das decisões tomadas, detalhando as funcionalidades e as tecnologias utilizadas para implementar o sistema.

3.1.1 Requisitos Funcionais

O presente trabalho tem os seguintes requisitos funcionais de sistema: Os requisitos funcionais são especificações das funções que o sistema deve ser capaz de realizar. Para um numero maximo *Skills* prováveis *N* definido *a priori*, estes incluem:

1. O sistema deve ser capaz de interpretar, pré processar e remover caracteres não legíveis de texto não estruturado, como CVs ou descrições de vagas de emprego, para garantir a qualidade dos dados de entrada.

2. Capacidade de reconhecer e extrair *Skills* em texto não estruturado de CVs ou descrições de vagas de emprego.

3. Capacidade de codificar cada *Skill* extraída num vetor de *embedding*.

4. O sistema deve ser capaz de fazer pesquisas vetoriais e semânticas na base de dados vetorial *Facebook AI Similarity Search* (FAISS) e devolver as *N* passagens textuais presentes com menor distancia vetorial, isto é, que tenham maior similaridade semântica com o texto pesquisado.

5. O sistema deve ser capaz de associar cada uma das *N* passagens textuais devolvidas pela pesquisa vetorial com uma entidade *Skill* na taxonomia ESCO.

6. Capacidade de atribuir uma pontuação (*score*) às *N* entidades *Skills* ESCO mais prováveis de ser corretamente atribuída à *Skill* extraída.

7. O sistema deve ser capaz de reconhecer a lista das *N Skills* ESCO prováveis de serem atribuídas para cada *Skill* extraída.

8. O sistema deve ser capaz de construir uma *prompt* especifica a cada *Skill* extraída contendo a própria *Skill* extraída do texto não estruturado, as entidades *Skill* na taxonomia ESCO, os exemplos textuais relevantes e a descrição ESCO para cada *Skill* provável.

9. O sistema deve ser capaz de selecionar apenas uma entidade *Skill* na taxonomia ESCO dentro da lista de *Skills* prováveis

3.1.2 Requisitos Não Funcionais

Os requisitos não funcionais descrevem as qualidades e restrições do sistema. Estes incluem:

1. O sistema atribui uma única entidade *Skill* da taxonomia ESCO por *Skill* extraída em cada inferência.

2. O sistema deve utilizar a *Graphics Processing Unit* (GPU) para aceleração por *hardware* durante o processo de inferência do LLM e na pesquisa vetorial na base de dados FAISS, garantindo alto desempenho.

3. O sistema aplica técnicas de quantização para otimizar o uso de memória durante a inferência do LLM, reduzindo a necessidade de recursos sem comprometer significativamente o desempenho.

4. O sistema deve garantir que não se baseia em viéses, crenças ou nuances textuais, tento a função de atribuição objetiva de *Skills* da taxonomia ESCO às *Skills* extraídas do texto.

3.1.3 Arquitetura Geral do Sistema SA4S-RH

Do ponto de vista arquitetural o trabalho é composto por:

- Modelo local de NLP *jjzha/escoxlmr_skill_extraction*, treinado a partir do modelo de NLP *XLM-Rlarge* (Zhang, 2024), que é usado na tarefa de NER para extrair a informação textual relativa a entidades "*Skill*".

- Modelo de NLP *all-mpnet-base-v2*, treinado para converter texto em vetores de *embedding*, que é usado para popular e fazer pesquisas na base de dados vetorial.

- Base de dados vetorial FAISS, que é usada para efetuar pesquisas vetoriais e receber os excertos textuais com menor distancia vetorial, correspondendo a maior similaridade textual.
- Dataset com as 13125 *Skills* da taxonomia ESCO, que contêm o índex da *Skill* ("idx"), o titulo da entidade ("title"), a descrição da entidade ("text"), o código ESCO associado ("esco_code") e o URL para a documentação da entidade ("documents_id").
- *Dataset* sintético com 10 exemplos gerados automaticamente para quase todas as *Skills* da taxonomia ESCO, que contêm o texto gerado ("sentence") e a respetiva *Skill* ESCO associada ("skill").
- O modelo LLM local baseado na arquitetura *Transformer*, *Llama2-7b* (Touvron et al., 2023), com 7 mil milhões de parâmetros treináveis.

3.1.4 Diagrama de Fluxo do Sistema SA4S-RH

A Figura 3.1 apresenta o diagrama de fluxo de alto nível do sistema proposto, detalhando as três fases de desenvolvimento do projeto e os componentes envolvidos em cada processo. O diagrama descreve sucintamente a sequência de operações permitindo uma visão completa do fluxo de trabalho.

No diagrama estão presentes 3 tipos de elementos: Entrada e saída de dados, representado por caixas obliquas; Processamento, sendo uma atividade que altera ou transforma fluxos de dados, representado por caixas retangulares e Acesso direto a bases de dados ou *datasets*, onde serão efetuadas atividades de pesquisa a bases de dados ou *datasets*, representado por caixas cilíndricas.

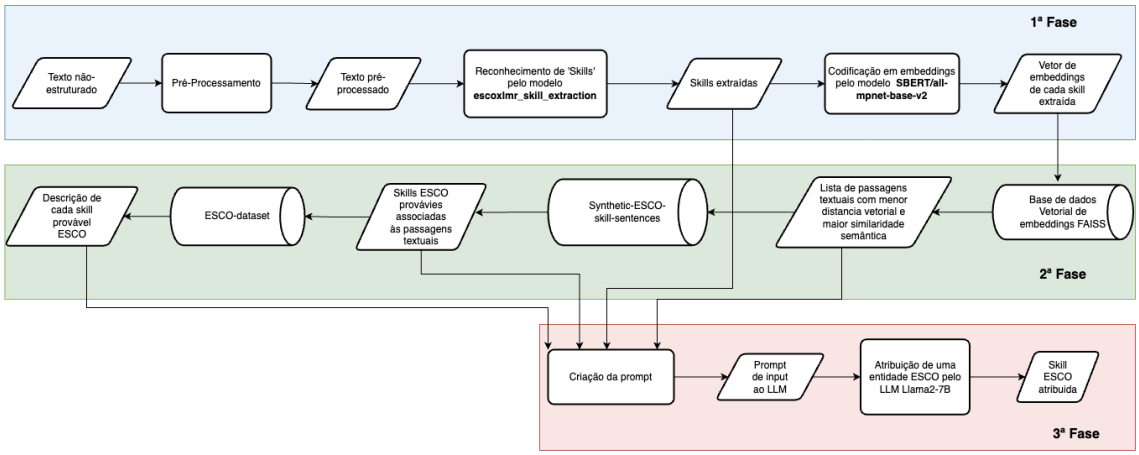


Figura 3.1: Diagrama de Fluxo de alto nível da Pipeline de Processamento do Sistema SA4S-RH

Fase 1: Extração e Conversão

A Figura 3.2 apresenta o diagrama de fluxo da primeira fase de **Extração e Conversão**, onde o texto não estruturado, proveniente dos CVs e descrições de vagas de emprego, serve como entrada para o sistema. Este texto passa por um processo de pré-processamento que remove caracteres especiais utilizados, sendo esta etapa inicial crucial para melhorar a precisão dos processos subsequentes.

Em seguida, o texto pré-processado é introduzido no modelo de NLP *escoxlmr_skill_-extraction*, desenvolvido para a tarefa de reconhecimento de entidades nomeadas (NER) focado na entidade "*Skill*". Como resultado deste processo, é devolvida uma lista de *Skills* extraídas do texto.

Por fim, para completar a primeira fase do sistema, a lista de *Skills* extraídas é convertida em vetores de *embedding* utilizando o modelo NLP SBERT/all-mpnet-base-v2, permitindo a continuação do processamento nas fases seguintes.

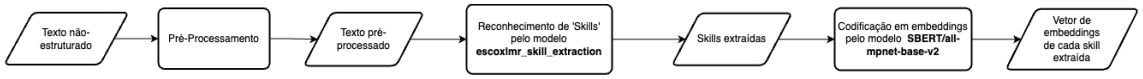


Figura 3.2: Diagrama de fluxo da primeira fase: Extração e Conversão

Fase 2: Calculo da distancia vetorial e pesquisa nas bases de dados

A Figura 3.3 apresenta o diagrama de fluxo da segunda fase de **Calculo da distancia vetorial e pesquisa nas bases de dados**, onde cada lista de vetores de *embedding*, resultantes da primeira fase e correspondente a cada uma das *Skills* extraídas, é usado como entrada para pesquisa na base de dados vetorial de *embeddings* FAISS. Como resultado do processo de acesso à base de dados é devolvida a lista de passagens textuais com menor distancia vetorial e maior similaridade textual e a corresponde pontuação (*score*) atribuída de 0 a 1, onde valores mais altos indicam maior similaridade.

Em seguida, cada elemento da lista de passagens textuais será utilizado como entrada para pesquisa na coluna "*sentence*" do *dataset Synthetic-ESCO-skill-sentences*, tendo como retorno o texto completo da passagem textual. Este passo é crucial para reconstruir a frase completa à qual a passagem textual pertence, sendo possível porque a base de dados vetorial FAISS foi treinada com os dados textuais da coluna "*sentence*" do *dataset*, garantindo que as passagens textuais retornadas estarão sempre presentes no próprio *dataset*.

As entidades *Skill* da ESCO são também recuperadas para cada passagem textual, consultando a coluna "*skill*" do *dataset*. A pontuação de similaridade semântica, atribuída pela base de dados vetorial, é utilizada como medida de probabilidade da *Skill* ser a correta.

São assim gerados dois dicionários: o primeiro contendo o texto completo das passagens textuais relevantes, correspondentes às entradas do *dataset Synthetic-ESCO-skill-sentences*, juntamente com as respetivas *Skills* da ESCO; o segundo com a lista das *Skills* da ESCO com maior probabilidade de serem atribuídas à *skill* extraída na **primeira fase** da *pipeline*, juntamente com as suas respetivas pontuações.

No passo final da segunda fase, cada uma das *Skills* mais prováveis é utilizada para consultar o *ESCO-dataset*, que contém todas as *Skills* da ESCO e as respetivas informações associadas. Esse processo permite recuperar a descrição oficial da ESCO correspondente a cada *Skill*.

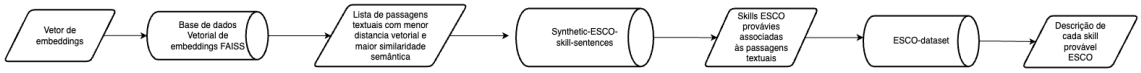


Figura 3.3: Diagrama de fluxo da segunda fase: Calculo da distancia vetorial e pesquisa nas bases de dados

Fase 3: Criação da prompt e tomada de decisão pelo LLM

A Figura 3.4 apresenta o diagrama de fluxo da terceira fase de **Criação da *prompt* e tomada de decisão pelo LLM**, onde será utilizada toda a informação recolhida na **primeira** e **segunda fases** para a construção da *prompt* que servirá como entrada para o LLM *Llama2-7B*.

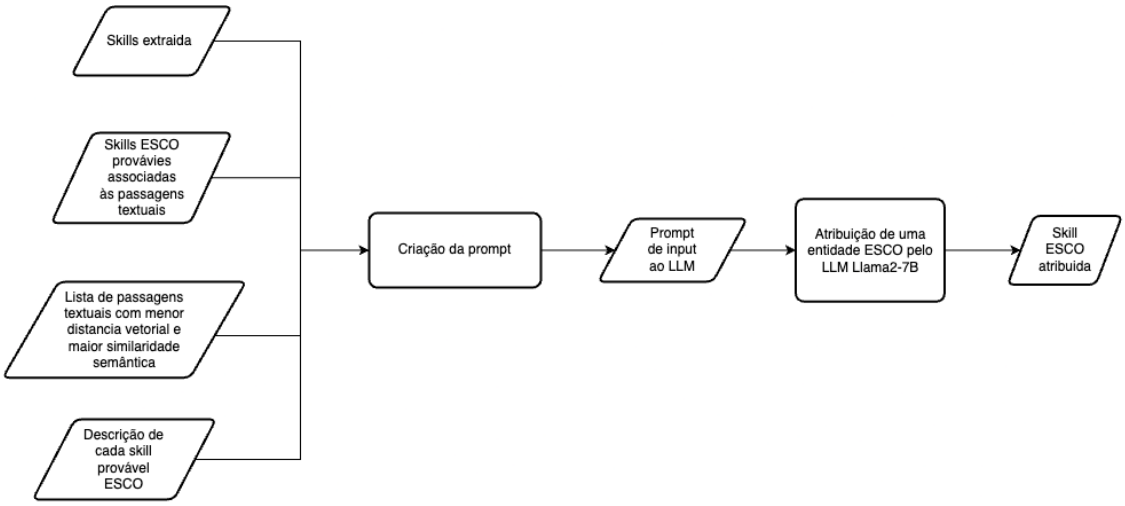


Figura 3.4: Diagrama de fluxo da terceira fase: Criação da prompt e tomada de decisão pelo LLM

A informação utilizada para a construção da *prompt* incluirá a *Skill* extraída do texto não estruturado proveniente da **primeira fase**, juntamente com as frases completas recuperadas na **segunda fase** e as respetivas *Skills* ESCO associadas, que servirão como exemplos para o LLM. Além disso, incluirá a lista das *Skills* ESCO e as suas respetivas

pontuações, indicando a probabilidade de serem corretamente atribuídas, também oriundas da **segunda fase**.

Este processo tem como objetivo explorar as capacidades do LLM através da engenharia de *prompt*, utilizando técnicas de *in-context learning* e *zero-shot learning*, para selecionar a entidade ESCO que melhor se adequa entre as opções mais prováveis selecionadas.

Este procedimento será repetido para cada uma das *Skills* extraídas do texto pré-processado na **primeira fase**. No final desta fase, todas as *Skills* extraídas terão uma *Skill* ESCO atribuída pelo LLM, completando assim o sistema.

3.1.5 Exemplo de Funcionamento do Sistema SA4S-RH

A Figura 3.5 demonstra o fluxo operacional da primeira fase do Sistema *Sistema Assistido para Seleção de Recursos Humanos* (SA4S-RH) através de um caso prático. Considere-se como entrada o texto: *"We need an employee who is able to assist our customers with their sewing patterns requests while providing excellent customer service experience."*. Este enunciado, representativo de uma vaga de trabalho, é depois de pré processado, submetido ao processamento pelo modelo de NER *escoxlmr_skill_extraction* para que sejam identificadas as entidades *Skill* com relação a entidades na taxonomia da ESCO, tendo sido detetadas no texto as menções a entidades *"assist customers with sewing patterns requests"* e *"providing excellent customer service experience"*. De seguida, cada uma destas entidades é processada pelo modelo de *embeddings SBERT/all-mpnet-base-v2* para converter o texto das entidades identificadas em vetores de *embedding*.

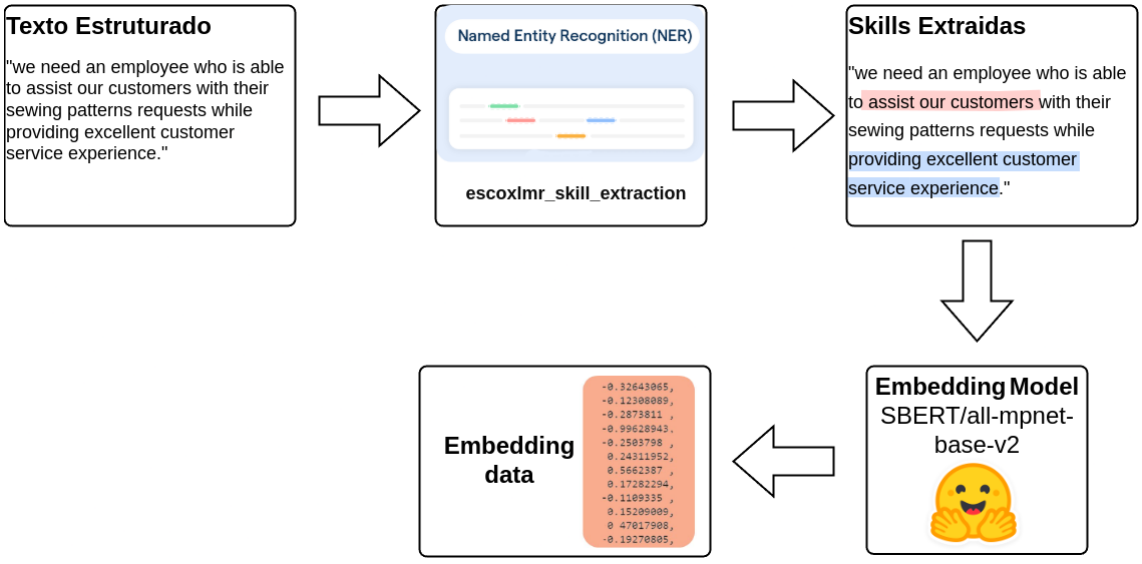
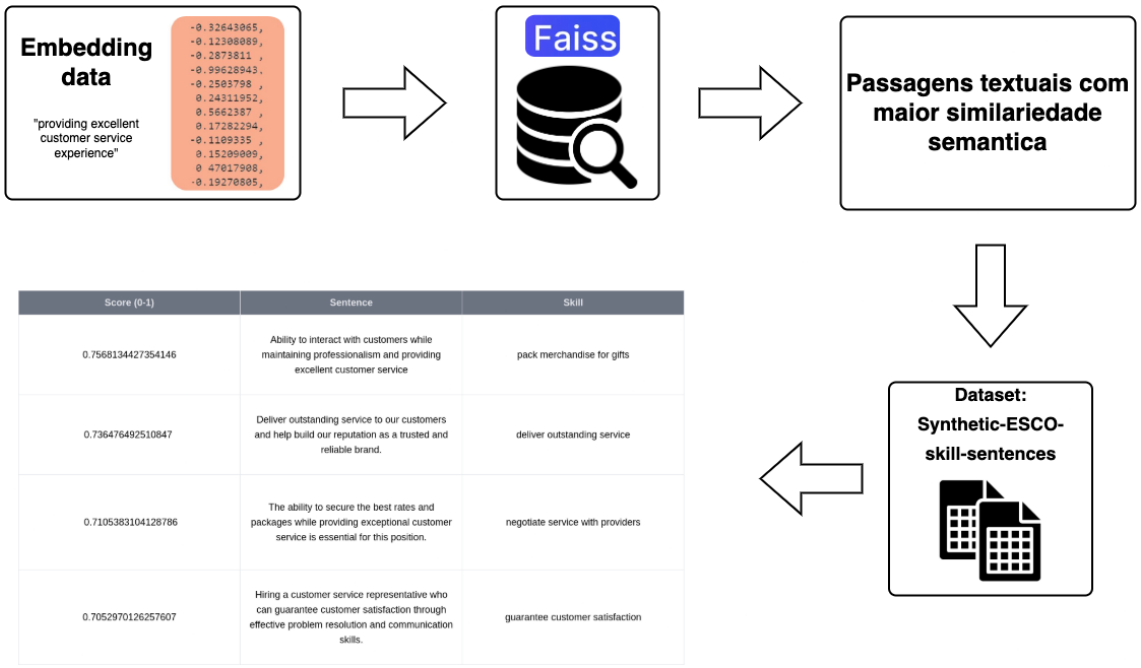


Figura 3.5: Exemplo da aplicação do Sistema SA4S-RH na primeira fase.

A Figura 3.6, ilustra o fluxo processual da segunda fase do sistema, que opera sobre os vetores de *embedding* relativos ao texto das entidades *Skill* identificativas, provenientes

da primeira fase. Estes são usado como pesquisa na base de dados vetorial FAISS, onde serão comparados os vetores de *embeddings* da entrada com os presentes na base de dados através do calculo da distancia euclidiana (L2) para encontrar as passagens textuais com menor distancia vetorial. Posteriormente, efetua-se uma consulta ao *dataset Synthetic-ESCO-skill-sentences* para recuperar o texto completo das passagens textuais devolvidas pela base de dados, assim como a *Skill* ESCO associada a cada uma dessas passagens. É ainda atribuída pela base de dados uma pontuação relativa à similaridade semântica de 0 a 1 entre o texto de entrada e a passagem textual devolvida, sendo essa informação também guardada. É possível encontrar na tabela da figura 3.6 a representação do retorno da segunda fase, que inclui a pontuação de similaridade, a frase completa recuperada e a *Skill* ESCO associada a essa frase. Nesta fase são assim definidas as entidades na taxonomia ESCO com probabilidade de ser atribuídas à entidade *Skill* extraída e a probabilidade é associada à pontuação de similaridade proveniente da pesquisa vetorial, assim como o exemplo relativo a essa entidade ESCO. As *Skills* ESCO prováveis são: *pack merchandise for gifts* com pontuação de cerca de 0.75, *deliver outstanding service* com pontuação de cerca de 0.73, *negotiate service with providers* com pontuação de cerca de 0.71 e *guarantee customer satisfaction* com pontuação de cerca de 0.70.



Score (0-1)	Sentence	Skill
0.7568134427354146	Ability to interact with customers while maintaining professionalism and providing excellent customer service	pack merchandise for gifts
0.736476492510847	Deliver outstanding service to our customers and help build our reputation as a trusted and reliable brand.	deliver outstanding service
0.7105383104128786	The ability to secure the best rates and packages while providing exceptional customer service is essential for this position.	negotiate service with providers
0.7052970126257607	Hiring a customer service representative who can guarantee customer satisfaction through effective problem resolution and communication skills.	guarantee customer satisfaction

Figura 3.6: Exemplo da aplicação do Sistema SA4S-RH na segunda fase, parte 1.

Na figura 3.7, ainda relativa à segunda fase é ilustrado o processo seguinte onde para cada uma das *Skills* prováveis será consultado o *dataset Use-ESCO-data* para recuperar a sua descrição oficial da ESCO. É possível encontrar na tabela a representação da entidade ESCO usado na consulta do *dataset* e a descrição de retorno.

Na Figura 3.8, correspondente à terceira fase da aplicação do sistema, é ilustrado o processo de construção do *prompt* a ser submetido ao LLM *Llama2*. Para esse efeito,

Figura 3.7: Exemplo da aplicação do Sistema SA4S-RH na segunda fase, parte 2.

recorreu-se aos resultados obtidos nas primeira e segunda fases, nomeadamente às *Skills* ESCO prováveis com a respetiva pontuação atribuída, às descrições ESCO de cada entidade e ao exemplo textual correspondente. Estes dados encontram-se apresentados na tabela da Figura 3.8. Assim, através de *prompt engineering* é construida a *prompt* a ser feita ao LLM para que este desempenhe a tarefa de EL e atribua entre as entidades ESCO provaveis a que melhor se adequa à entidade *Skill* extraída na primeira fase. No final do processo o LLM *Llama2* atribuiu à *Skill* extraída do texto "*providing excellent customer service experience*"a *Skill* ESCO "*Deliver outstanding service*". O sistema prosseguirá, iterando para a outra *Skill* extraída "*assist customers with sewing patterns requests*", completando assim o ciclo de operações.

3.1.6 Tecnologias Usadas

Com base nos requisitos identificados, foram selecionadas as seguintes tecnologias para o desenvolvimento do sistema:

Linguagens de programação

A linguagem de programação *Python* foi escolhida para o desenvolvimento do projeto devido ao seu amplo suporte para bibliotecas essenciais, como a *Transformers*, que é crucial para a utilização de modelos baseados na arquitetura *Transformers*, como os modelos *RoBERTa* e o *LLaMA2*. *Python* é a linguagem padrão para bibliotecas de ML e AI, o que justifica a sua escolha, pois oferece robustez e flexibilidade necessárias para o desenvolvimento de soluções avançadas em aprendizado de máquina e inteligência artificial.

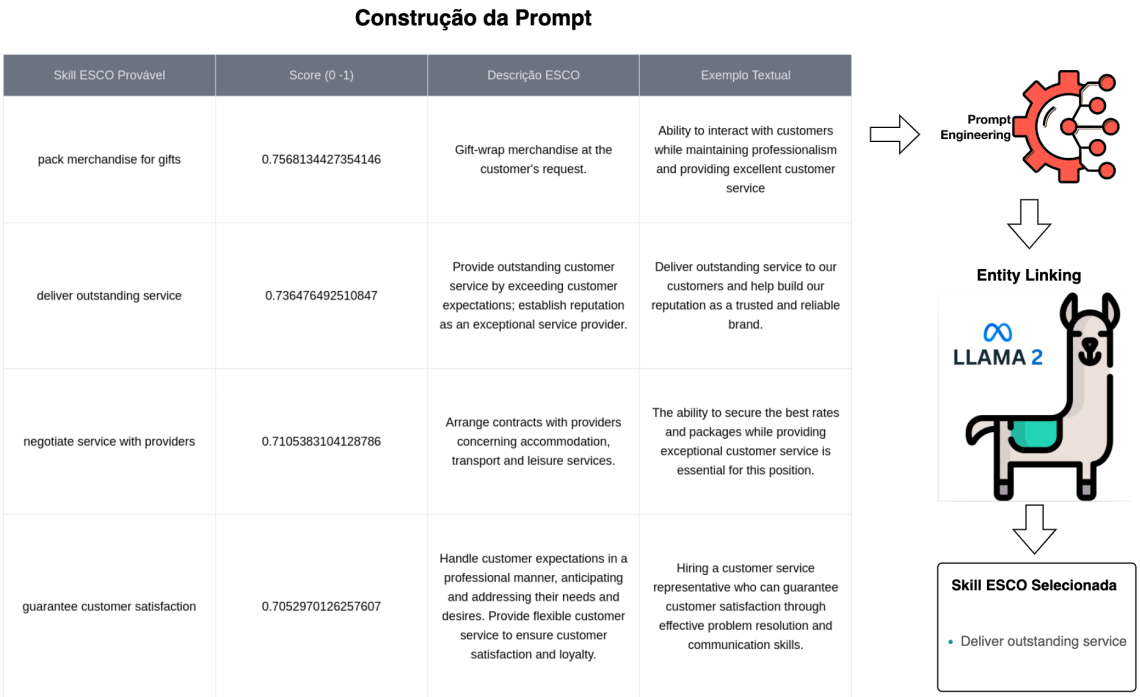


Figura 3.8: Exemplo da aplicação do Sistema SA4S-RH na terceira fase.

Ambiente de Desenvolvimento

Para o ambiente de desenvolvimento, incluindo o servidor em nuvem para hospedagem dos recursos de máquina, foram utilizados o *Google Colab* (conta gratuita) e o *Google Cloud Platform*. Essas plataformas permitem a criação de *kernels* que correspondem a ambientes virtuais de *Python* já configurados com bibliotecas essenciais pré-instaladas. Essa escolha facilita o desenvolvimento e a execução de código, proporcionando um ambiente eficiente e acessível para o projeto, assim como para hospedagem dos modelos de ML. (da Silva Ferreira et al., 2023)

Bibliotecas Python

Foram usadas as respetivas bibliotecas *Python*, nas seguintes versões:

1. **accelerate==0.34.2**
2. **bitsandbytes==0.44.1**
3. **einops==0.8.0**
4. **faiss-gpu==1.7.1.post3**
5. **langchain==0.1.4**
6. **pandas==2.1.4**

7. **pathlib==1.0.1**
8. **pypdf==5.0.1**
9. **sentence_transformers==3.1.1**
10. **torch==2.4.1**
11. **tokenizers==0.19.1**
12. **transformers==4.44.2**
13. **xformers==0.0.22.post7**

Plataforma de Repositório de modelos de NLP

A plataforma escolhida para repositório do modelo de NLP foi a *Hugging Face*, devido à sua infraestrutura poderosa, compatibilidade com serviços de nuvem de nível empresarial, vasta biblioteca de modelos pré-treinados e suporte a uma comunidade ativa de investigadores e profissionais na area de AI.

A *Hugging Face* destaca-se pela sua biblioteca *Transformers* (Wolf et al., 2020), amplamente reconhecida e utilizada para tarefas complexas de NLP, como classificação de texto, reconhecimento de entidades, tradução automática, entre outras. Esta biblioteca de código publico e aberto oferece uma vasta gama de modelos pré-treinados, incluindo *BERT*, *GPT*, entre outros, o que facilita a utilização de modelos de ML de última geração sem a necessidade de começar o desenvolvimento do zero.

Além disso, a *Hugging Face* possui uma infraestrutura robusta que permite o fácil alojamento e promove uma forte comunidade de software de código publico e aberto, permitindo a partilha de modelos, *datasets* e a colaboração em projetos de ML.

Modelo de Embedding

A escolha do modelo de *embeddings all-mpnet-base-v2* foi fundamentada pela sua elevada qualidade no desempenho de tarefas de representação em codificação de *embeddings* para frases ou pequenos parágrafos, tornando-o adequado para as necessidades deste projeto. Este modelo foi treinado em mais de mil milhões de pares de frases utilizando a metodologia de treino por aprendizagem auto-supervisionada contrastiva (*self-supervised contrastive learning objective*), o que garante a excelente capacidade de capturar a semântica das frases em vetores de *embedding*, podendo estes depois ser usados em diversas tarefas de NLP, como *Retrieval-Augmented Generation* (RAG), pesquisa por similaridade semântica ou em tarefas de *text clustering* (tarefa de NLP que envolve o agrupamento de documentos similares baseado no seu conteúdo textual, identificando padrões e tendências).

A escolha foi feita também pelo suporte da biblioteca *Sentence Transformers*, amplamente utilizada e fácil de integrar, permitindo gerar vetores de alta qualidade que podem ser usados de forma eficiente (Reimers and Gurevych, 2019), conforme as necessidades deste trabalho. O modelo encontra-se disponível no repositório *Hugging Face* (**URL** `https://huggingface.co/sentence-transformers/all-mpnet-base-v2`).

Base de Dados Vetorial

FAISS foi a base de dados vetorial escolhida para este projeto. O FAISS é uma biblioteca que permite a criação de bases de dados vetoriais para pesquisa de similaridade eficiente e agrupamento de vetores densos. Contém algoritmos que permitem a pesquisa em conjuntos de vetores de qualquer tamanho, até aqueles que possivelmente não cabem na *Random-access Memory* (RAM). A biblioteca também inclui código de suporte para avaliação e ajuste de parâmetros. Desenvolvido em *C++* com suporte completo para *Python*, alguns dos algoritmos mais úteis são implementados para ser usados de forma eficiente na GPU (Johnson et al., 2019).

FAISS contém diversos métodos para pesquisa por similaridade semântica, sendo as instâncias representadas como vetores e identificadas por um número inteiro podendo ser comparados por distâncias Euclidianas (L2) ou produtos escalares. Vetores que são similares a um vetor de consulta são aqueles que têm a menor distância L2 ou o maior produto escalar com o vetor de consulta. FAISS também suporta similaridade cosseno, sendo equivalente ao produto escalar em vetores normalizados (Johnson et al., 2019).

A implementação na GPU pode aceitar entradas tanto da memória da *Central Processing Unit* (CPU) quanto da GPU, obtendo resultados das pesquisa mais rápido, se tanto a entrada quanto a saída permanecerem residentes na GPU (Johnson et al., 2019). A escolha do FAISS em detrimento de outras bases de dados vetoriais foi motivada por este oferecer uma combinação de eficiência e flexibilidade, sendo capaz de lidar com conjuntos de dados de grande escala que podem não caber na memoria RAM. Além disso, a sua implementação eficiente em GPU permite um desempenho superior, essencial para tarefas que requerem pesquisas por similaridade rápida e precisa em vetores de grande dimensão.

Outro fator decisivo é o suporte extensivo que o FAISS fornece para a avaliação e ajuste de parâmetros, facilitando o desenvolvimento e a otimização do sistema. Comparado com outras alternativas, o FAISS apresenta uma comunidade de desenvolvimento ativa e é amplamente utilizado em pesquisas e aplicações industriais, o que garante uma base sólida de suporte e documentação. A integração completa com *Python* também foi um critério importante, pois facilita a utilização e implementação dos modelos dentro do ecossistema de desenvolvimento existente.

Dessa forma, a base de dados FAISS atende às necessidades técnicas do projeto e oferece uma solução robusta e escalável para o processamento e procura de vetores densos,

justificando a sua escolha como a base de dados vetorial a ser usada.

Modelo de NLP para a tarefa de NER

O modelo de NLP escolhido para a tarefa de NER, de extração de entidade *Skill* foi o modelo *jjzha/escoxlmr_skill_extraction*, disponível no repositório da *Hugging Face*, (**URL** `https://huggingface.co/jjzha/escoxlmr_skill_extraction`).

O modelo desenvolvido por Zhang, no âmbito da sua tese de doutoramento, baseia-se no *XLM-Rlarge* (Zhang, 2024) e é adaptado especificamente para tarefas relacionadas com o mercado de trabalho, utilizando o vocabulário e a taxonomia ESCO. Este modelo combina técnicas de pré-treino orientado ao domínio com objetivos auto-supervisionados, como o MLM e o *ESCO Relation Prediction* (ERP), otimizando a sua capacidade de identificar e relacionar ocupações e competências em múltiplos idiomas, o que melhora significativamente a precisão em tarefas de NER para a extração de habilidades profissionais.

Modelo de Linguagem

O modelo LLM escolhido foi o *Llama2-7B*, desenvolvido pela *Meta AI* (Touvron et al., 2023). Este modelo de código publico e aberto (*open source*) faz parte da família de modelos pré-treinados e *finetuned*, o *Llama 2* e o *Llama 2-Chat*, com escalas de 7, 13 e 70 mil milhões de parâmetros, tendo sido escolhido o modelo de 7 mil milhões de parâmetros por ser mais ajustado aos recursos maquina disponíveis. Nos testes de performance realizados, o LLM *Llama2* apresenta um desempenho superior aos modelos de *open source* existentes, até então. Este modelo consegue ser comparado a alguns dos modelos de código fechado, em algumas das avaliações humanas realizadas (Touvron et al., 2023).

A escolha do *Llama2-7B* foi motivada pelas capacidades avançadas de entendimento e geração de texto em linguagem natural, bem como pelo compromisso com a segurança e a transparência no desenvolvimento do modelo. Uma vez que é *open source*, este consegue ser hospedados localmente, não sendo necessária a utilização de *Application Programming Interface*s (APIs) pagas fornecidas por LLMs de código privado e fechado (*closed source*), como os da *Openai* (Touvron et al., 2023).

Datasets Usados

Synthetic-ESCO-skill-sentences

Para a criação da base de dados vetorial foi usado o *dataset Synthetic-ESCO-skill-sentences* de Decorte et al.. Este *dataset* é composto por anúncios de emprego sintéticos gerados por um LLM para cada *Skill* ESCO (Decorte et al., 2023). Este contém 10 frases de anúncios de emprego sinteticamente geradas para quase todas (99.5%) as *Skills* presentes na versão 1.1.0 do ESCO, em inglês.

O *dataset* é composto por 138.260 pares frase (coluna "*sentence*") e habilidade (coluna "*skill*"). (**URL** `https://huggingface.co/datasets/jensjorisdecorte/Synthetic-ESCO-skill-sentences`) Na tabela seguinte é mostrado a tabela com exemplos do *dataset Synthetic-ESCO-skill-sentences*:

Sentence	**Skill**
The ideal candidate for this position should be able to advise customers on sewing patterns based on their needs.	Advise customers on sewing patterns
Experience with environmental, sustainability, social, and financial matters as they relate to urban planning law is a plus.	Urban planning law
If you possess expertise in the area of procurement market analysis, we would like to hear from you.	Perform procurement market analysis

Tabela 3.1: Exemplos do *dataset Synthetic-ESCO-skill-sentences*.

Este *dataset* desempenhou um papel crucial na criação da base de dados vetorial e na validação do modelo proposto. Este forneceu uma ampla variedade de exemplos que permitiram a construção de uma base vetorial rica em informações sobre *Skills* profissionais da taxonomia ESCO, oriundas de diferentes contextos. A utilização deste *dataset* foi fundamental para garantir que o modelo fosse capaz de lidar com uma grande diversidade de *Skills* e em diferentes contextos, o que aprimorou significativamente sua capacidade de generalização e precisão na tarefa de filtro das entidades ESCO provaveis.

Use-ESCO-data

O *dataset Use-ESCO-data* foi disponibilizado pela ESCO através do seu site oficial (**URL** `https://esco.ec.europa.eu/en/use-esco/download`). A plataforma permite o download gratuito do *dataset* em todas as 28 línguas disponíveis na ESCO, tendo sido usada a versão 1.1.0 em inglês. O *dataset* contem para todas as *Skills* ESCO, o respetivo titulo ("*title*"), a descrição oficial ("*text*"), o código ESCO da *Skill* ("*esco_code*") e o URL da *Skill* ("*document_id*").

3.1.7 Diagrama de Arquitetura do Sistema SA4S-RH no Colab

Todos os elementos da *pipeline*, incluindo os modelos, *datasets* e a base de dados vetorial FAISS, foram alocados num *kernel* do *Colab*, correspondendo a um ambiente virtual de *Python*. A versão gratuita do *Colab* disponibiliza recursos de *hardware*, incluindo a GPU *NVIDIA T4* com 16 GB de *RAM*.

O diagrama da Figura 3.9 exibe todos os componentes da *pipeline* e a sua alocação nos recursos disponibilizados pelo *Colab*. Entre os modelos de NLP alocados na GPU *T4* para aceleração por *hardware*, encontram-se o modelo de NER *escoxlmr_skill_extraction*, o modelo de *embeddings sentence-transformers/all-mpnet-base-v2*, o LLM *Llama2-7B* e

a base de dados vetorial FAISS. Esta configuração permite a otimização dos processos de pesquisa e inferência, ao partilhar a memória disponível da GPU e a aproveitar as capacidades avançadas de processamento paralelo proporcionadas pela GPU com suporte *CUDA*. Os *datasets Synthetic-ESCO-skill-sentences* e *Use-ESCO-data*, utilizados nos processos da *pipeline*, foram alocados diretamente na *RAM* do ambiente virtual do *Colab*.

No diagrama, as cores representam a organização dos elementos da arquitetura, a área a amarelo indica os modelos de NLP fornecidos e disponíveis no repositório da *Hugging Face*, a área a azul identifica os componentes desenvolvidos pela *Meta AI*, e a área em roxo representa os *datasets* utilizados.

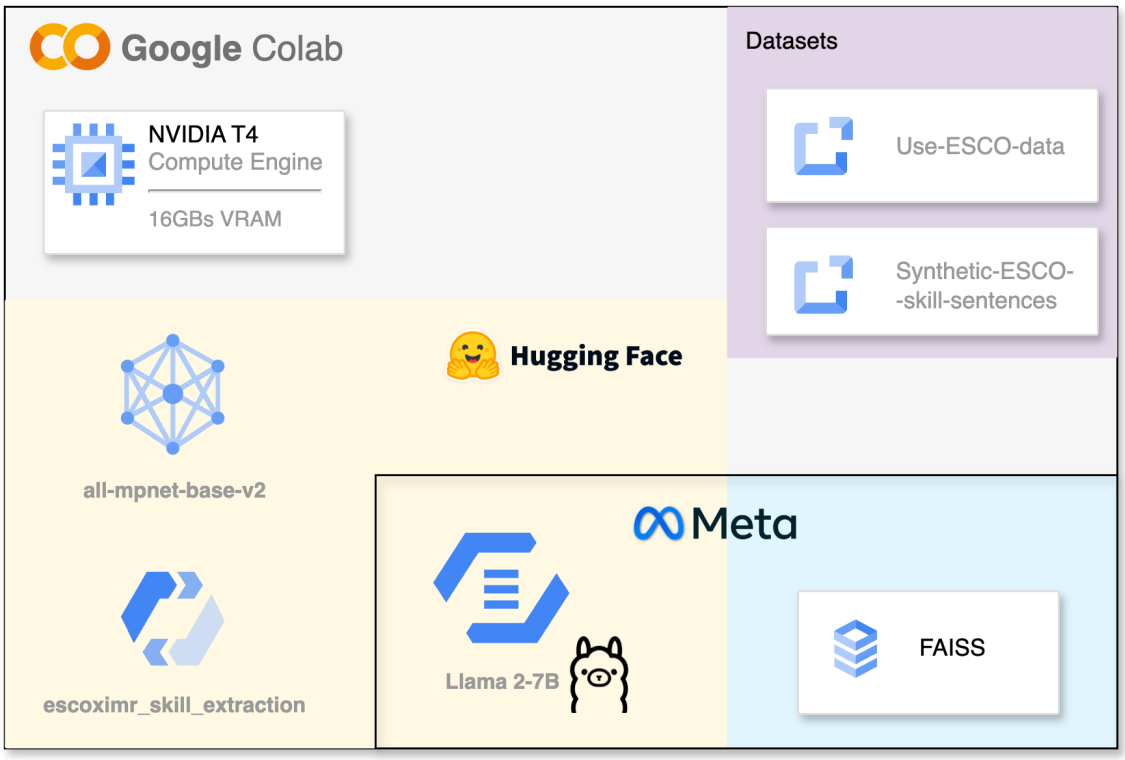


Figura 3.9: Diagrama de Arquitetura do Sistema SA4S-RH no *Colab*

3.2 Implementação do sistema SA4S-RH

O presente trabalho propõe uma abordagem inovadora, denominada de SA4S-RH, com objetivo de utilizar as capacidades dos LLM em tarefas de NLP e *Entity Linking* (EL), para atribuir uma entidade *Skill* na taxonomia ESCO a uma *Skill* extraída de texto não estruturado de descrições de vagas de emprego e CVs. O sistema SA4S-RH serve-se da capacidades dos LLMs para aprendizagem em contexto (*in-context learning*) na abordagem de *Zero-shot learning* (ZSL) para que não seja necessário *finetuning* do modelo e que este consiga realizar a tarefa de EL.

Este trabalho suporta-se no estado da arte atual dos LLM e nos trabalhos relacionados desenvolvido na area dos LLM para tarefas de NLP, nomeadamente em tarefas de NER e EL. Tendo em conta as provadas capacidades dos LLM em tarefas de NER e EL, sem recurso a pré-treino ou *finetuning*, leva a que estes sejam facilmente adaptáveis a contextos específicos através da utilização de técnicas de ZSL e *Few-Shot Learning* (FSL), sendo necessário adaptar a *prompt* a ser feita ao LLM (Ding et al., 2024; Zhang et al., 2024b).

No entanto, como observado, as principais limitações da utilização de LLMs em tarefas de NER e EL estão relacionadas à eficiência e performance quando o número de entidades cresce além de algumas dezenas. Portanto, a utilização isolada dos LLMs não seria suficiente para atender às necessidades do projeto, uma vez que a tarefa de atribuição de entidades ESCO a cada *Skill* deve considerar todas as 13125 entidades ESCO (Ding et al., 2024).

Para enfrentar esse desafio, este trabalho propõe a utilização de uma base de dados vetorial com objetivo de filtrar e limitar o número de entidades ESCO com potencial de serem associadas à *Skill* extraída. Para isso, utiliza-se a base de dados FAISS populada com vetores de *embedding* provenientes da coluna "*sentence*" do *dataset Synthetic-ESCO-skill-sentences*. Nesse *dataset*, cada entidade ESCO possui 10 exemplos textuais gerados sinteticamente por um LLM, sendo utilizados apenas os exemplos que não fazem menção explícita à entidade ESCO correspondente.

O objetivo deste processo é criar uma base de dados vetorial com vetores de *embedding* diretamente relacionados ao contexto do mercado de trabalho na taxonomia ESCO, onde seja possível fazer pesquisas pela distancia euclidiana (L2), correspondendo a similaridade semântica, entre os vetores de *embedding* presentes na base de dados vetorial e o vetor de *embedding* da *Skill* extraída do texto dos CVs ou descrição da vaga de emprego. Após a obtenção das N passagens textuais com menor distância L2, correspondentes à maior similaridade semântica resultante da pesquisa na base de dados FAISS, é possível identificar as *Skills* ESCO associadas por meio da consulta à coluna "*skill*" no *dataset Synthetic-ESCO-skill-sentences*. Dessa forma, com as referências fornecidas pela base de dados, é possível consultar o *dataset* para recuperar o texto completo da passagem textual e a entidade ESCO correspondente a cada uma das N passagens, onde N é o número de passagens definido na pesquisa. A pontuação (*score*) atribuída pela distância L2 a cada uma das N passagens textuais é também armazenada, servindo como uma probabilidade associada a cada uma das possíveis *Skills* ESCO, variando de 0 a 1, onde valores mais altos indicam maior similaridade semântica.

Sendo a base de dados FAISS criada e otimizada para o contexto do mercado de trabalho, contendo 10 exemplos para cada um das 13125 *Skills* ESCO, é ideal para ser usada como método de filtragem das *Skills* ESCO prováveis de corresponder à *Skill* ESCO mais apropriada a ser atribuída à *Skill* extraída do texto não estruturado.

O passo seguinte da metodologia foi utilizar as capacidades dos LLMs em tarefas

de NLP, nomeadamente de EL, para selecionar entre as principais entidades ESCO com maior probabilidade, a que melhor se adequa à *Skill* que foi extraída. Neste processo de decisão e EL por parte do LLM será necessário utilizar engenharia de *prompts* para guiar o LLM *Llama2-7B* a aprender por contexto (*in-context learning*) na abordagem de ZSL, ou seja, sem que sejam fornecidos exemplos prévios. Dessa forma, a *prompt* será construída a partir de um *template* predefinido, que incluirá a instrução para o modelo sobre a tarefa a ser realizada e o comportamento esperado, a *Skill* extraída do texto, as principais entidades ESCO prováveis com suas respetivas pontuações (*score*) de 0 a 1, os exemplos textuais completos de cada *Skill* identificados na pesquisa vetorial, e a descrição oficial ESCO de cada entidade provável. Esta abordagem garante que as vantagens do uso dos LLM nas tarefas de EL sejam aproveitadas, já que o número de entidades prováveis é limitado no máximo a *N*, sendo *N* representa um número inteiro. No caso da base de dados selecionar entidades repetidas, pertencentes à lista das *N Skills* prováveis, será mantida a entidade, que receberá a maior pontuação encontrada entre as duplicadas.

O resultado devolvido pelo LLM será a *Skill* ESCO, ou no caso de nenhuma entidade se enquadrar à *Skill* extraída, é devolvido um valor simbólico de ausência de *Skill* ("*None*").

3.2.1 Prompt usada

Foi utilizada a seguinte *prompt* para o LLM, seguindo as recomendações de *prompt* do *LLama2*. É importante distinguir entre a instrução dada ao modelo e a *prompt*. A instrução orienta o modelo sobre como deve comportar-se ou qual é o contexto geral da tarefa, enquanto a *prompt* especifica a tarefa ou pergunta exata que deve ser respondida pelo modelo.

A estrutura adotada, seguindo as recomendações de estrutura para o modelo *LLama2*, foi a seguinte:

```
[INST] <<SYS>>
{Instrução a ser dada ao modelo}
<</SYS>>
{Prompt/Pergunta a ser feita ao modelo}
[/INST]
```

A instrução escolhida para ser fornecida ao modelo foi a seguinte. Esta encontra-se em inglês uma vez que a maior parte dos dados e instruções realizadas durante o processo de treino do LLM *Llama2* encontram-se na língua inglesa.

```
[INST] <<SYS>>
You are a helpful assistant. Your job is to choose the right skill in
the ESCO taxonomy for the given text. Use the context information to
```

```
help you choose the right ESCO entity of the skill. Each provable ESCO
entity includes the rank (from 0 to 1). Only output the
chosen ESCO entity.
<</SYS>>
Given text to link to a ESCO entity:
{extracted_skill}

Most probable ESCO entities for the given text are:
{probable_entities}

Official Description for each ESCO entity:
{esco_skill_description_text}

Examples of the right ESCO entities for a given text to help you choose
the right ESCO entity:
{esco_examples}

Select the right entity from the following options. If none of the
entities seem appropriate, return None.
[/INST]
```

3.2.2 Extração de entidades Skill

A tarefa de NLP para extração de entidades (NER) "*Skill*" do texto não estruturado das descrições de vagas de emprego ou CVs é realizada através da utilização do modelo de NLP *jjzha/escoxlmr_skill_extraction* derivado do modelo *ESCOXLM-R* criado por Zhang et al.. Este modelo foi baseado no modelo de NLP *XLM-Rlarge*, um modelo multilíngua baseado na arquitetura *transformers*, concebido para lidar com os requisitos específicos de tarefas relacionadas com o trabalho em vários idiomas (Zhang et al., 2023; Zhang, 2024).

Extrair *Skills* de texto não estruturado utilizando o modelo *jjzha/escoxlmr_skill_extraction* envolve vários passos importantes. Quando um excerto de texto é introduzido no modelo este identifica e classifica *tokens* que provavelmente representam competências. Cada *token* identificado como *Skill* recebe um rótulo de grupo de entidades, como "B" (início de uma *Skill*) ou "I" (dentro de uma *Skill*), juntamente com uma pontuação de confiança que reflete a probabilidade de o *token* pertencer a um determinado grupo de entidades. O modelo gera também a posição do *token* dentro do texto (índices inicial e final) e a *Skill* extraída correspondente (Zhang et al., 2023; Zhang, 2024).

Os resultados são depois processados para garantir que *tokens* consecutivos que per-

tencem à mesma entidade extraída (ou seja, partes da mesma entidade *Skill*) são agrupados. Por exemplo, se o modelo identificar "aconselhar" como o início de uma *Skill* e depois "clientes sobre padrões de costura" como a continuação da *Skill*, estes *tokens* serão combinados numa única entidade, representando a *Skill* completa.

Na prática, o modelo processa cada entidade detetada no texto, e se os *tokens* forem consecutivos e fizerem parte da mesma *Skill*, são fundidos para formar uma *Skill* coerente e completa. Este processo é essencial para combinar *tokens* individuais que podem ter sido fragmentados em várias faixas numa única frase de *Skill* significativa.

O resultado final inclui as competências retiradas do texto, as suas posições (índices inicial e final) e os respetivos índices de confiança. Este método contribui para que as competências sejam identificadas e reconstruidas com precisão, permitindo que o sistema lide eficazmente com entradas complexas ou fragmentadas.

Exemplo de retorno do modelo de NER

A Figura 3.10 apresenta um exemplo de retorno do modelo de NER utilizado para extração de *Skills*. O modelo é capaz de identificar o início (marcado como "B", de *Begin*) e a continuação (marcada como "I", de *Inside*) de uma *Skill* ao processar um texto. Além disso, o modelo atribui uma pontuação de confiança (*score*) a cada *token* extraído, indicando a probabilidade de que o *token* pertença à entidade identificada como *Skill*.

No exemplo da figura, o modelo extrai as *tokens* relacionadas a uma *Skill* do texto de entrada, retornando não apenas as palavras individuais que compõem a *Skill*, mas também a posição de início e término das palavras no texto original, o que permite a identificação precisa das *Skills* mencionadas.


```
[
  {'entity_group': 'B', 'score': 0.9996942, 'word': 'advise',
  'start': 56, 'end': 62, 'entity': 'Skill', 'extracted_tokens': 'advise'},
  {'entity_group': 'I', 'score': 0.99927, 'word': 'customersonsewingpatterns',
  'start': 63, 'end': 91, 'entity': 'Skill', 'extracted_tokens': 'customers on sewing patterns'},
  {'entity_group': 'I', 'score': 0.99950916, 'word': 'customersonsewingpatterns',
  'start': 120, 'end': 148, 'entity': 'Skill', 'extracted_tokens': 'customers on sewing patterns'}
]
```


Figura 3.10: Exemplo de retorno do modelo de NER

3.2.3 Criação da Base de Dados Vetorial

Para a criação da base de dados vetorial, utilizou-se a coluna "*sentence*" do *dataset Synthetic-ESCO-skill-sentences*. Utilizando a biblioteca FAISS, foi possível criar uma base de dados vetorial com a aplicação do modelo de *embeddings sentence-transformers/all-mpnet-base-v2*. Este modelo converte cada segmento de texto em vetores de *embedding*,

com cada segmento definido por um tamanho máximo de 1000 *tokens* (*chunk_size=1000*) e uma sobreposição de 20 *tokens* entre segmentos adjacentes (*chunk_overlap=20*).

O tamanho do pedaço ou *chunk (chunk size)* corresponde ao número máximo de caracteres que um segmento pode conter. A sobreposição de *chunks (chunk overlap)* é o número de caracteres que deve ser compartilhado entre dois segmentos adjacentes. Os parâmetros de tamanho e sobreposição dos *chunks* podem ser utilizados para controlar a granularidade da divisão do texto. Um tamanho de *chunk* menor resultará em mais segmentos, enquanto um tamanho de *chunk* maior resultará em menos segmentos. Uma maior sobreposição de *chunks* resultará em mais segmentos que partilham caracteres comuns, enquanto uma menor sobreposição resultará em menos segmentos que partilham caracteres comuns.

Esta base de dados vetorial tem como função principal procurar os *N* segmentos textuais mais semelhantes à consulta fornecida, sendo os parâmetros escolhidos de forma estratégica para equilibrar a granularidade da análise do texto e a eficiência do sistema. O tamanho de *chunk* a 1000 *tokens* foi definido para capturar adequadamente a informação sem segmentar excessivamente o texto, sendo que uma maior granularidade poderia prejudicar a relação semântica entre palavras que precisam de estar no mesmo contexto. A sobreposição de 20 *tokens* foi utilizada para assegurar que palavras ou frases importantes próximas ao limite dos *chunk* não fossem separadas, preservando a continuidade do significado entre segmentos.

Foi desenvolvida uma metodologia que, a partir de uma consulta de entrada, pesquisa na base de dados vetorial os *N* segmentos textuais mais relevantes. O resultado é devolvido como um dicionário em *Python*, contendo as frases completas (coluna "*sentence*" do *dataset Synthetic-ESCO-skill-sentences*) e as respetivas habilidades (coluna "*skill*" do *dataset Synthetic-ESCO-skill-sentences*).

A abordagem adotada permitiu a eficiente criação de uma base de dados vetorial que facilita a realização de pesquisas e análises de similaridade entre os textos correspondentes aos exemplos gerados a partir da descrição oficial das *Skills* ESCO e as *Skills* extraídas que serão usadas como consulta de entrada à base de dados. Estas *Skills* devolvidas pela utilização desta metodologia servirá como auxilio ao LLM na filtragem de entidades a que destinará a associação à entidade ESCO mais adequada (tarefa de EL).

3.2.4 Hospedagem local do LLM Llama2-7B

Para hospedar e otimizar o modelo LLM *Llama2-7B* para inferência eficiente num ambiente com recursos de *hardware* limitados, foi utilizado o *Google Colab*, que na versão gratuita disponibiliza a GPU *NVIDIA T4* com 16 GB de RAM. A GPU *T4* foi usada especificamente para realizar a inferência do LLM, acelerando significativamente o desempenho do modelo em tarefas de processamento de linguagem natural (NLP). O ambi-

ente foi configurado para utilizar a biblioteca *PyTorch*, com recurso à camada de acesso a instruções virtuais da GPU *CUDA*.

Quantização para eficiência de memoria

Para reduzir o consumo de memória do modelo foi aplicada a técnica de quantização, permitindo que o LLM fosse carregado na memória limitada da GPU (16 GB). A quantização em 4 *bits* foi empregue, sendo um método eficaz para minimizar o uso de memória sem comprometer de forma significativa o desempenho do modelo. A configuração de quantização foi realizada com a biblioteca *BitsAndBytes*, utilizando a técnica *NF4* para garantir uma maior eficiência computacional e menor consumo de memória.

Caraterísticas da quantização aplicada ao LLM:

- Quantização de 4 *bits*: Reduz a precisão dos pesos do modelo para 4 *bits*, diminuindo significativamente a quantidade de memória necessária para armazenar os parâmetros do modelo.
- Tipo de quantização *NF4*: Trata-se de uma técnica de quantização específica denominada de *normal float 4 (NF4)*, que ajuda a melhorar a estabilidade numérica dos pesos de 4 *bits*.
- Dupla Quantização: O modelo utiliza também a dupla quantização, onde é aplicada uma camada adicional de quantização em 4 *bits* para comprimir ainda mais o modelo e aumentar a eficiência computacional.
- *Bfloat16* para computação: O modelo utiliza *bfloat16* (*brain floating point*) para cálculos internos, oferecendo um bom equilíbrio entre precisão e desempenho em hardware moderno como a GPU *T4*.

Esta estratégia de quantização permite que o modelo *Llama 2-7B*, que normalmente requer substancialmente mais memória, se enquadre no limite de memória do GPU *T4* de 16 GB, mantendo uma inferência rápida e eficiente.

Inferência de modelo

O modelo é carregado utilizando a biblioteca *Transformers*, da *Hugging Face*, a partir do repositório oficial da *Meta*, com as definições de quantização especificadas. Em seguida, o modelo é configurado no modo de avaliação, o que otimiza o desempenho para inferência, em vez do modo de treino. O *tokenizer* também é inicializado para gerir o pré-processamento e o processo de transformação do texto de entrada em *tokens* (*tokenization*).

Critérios de paragem de inferência personalizados foram implementados para controlar o processo de geração de texto, evitando que a saída se prolongue excessivamente.

Estas configurações permitem que o modelo *Llama 2* realize tarefas de geração de texto de forma otimizada, eficiente e com alta qualidade, respeitando as limitações de recursos de *hardware* usados.

A temperatura do LLM, parâmetro que influencia o conteúdo gerado pelo modelo, foi ajustada para o valor mínimo de 0. Este parâmetro controla o equilíbrio entre criatividade e previsibilidade nas respostas, onde quanto maior a temperatura, maior a variabilidade e a criatividade nas saídas. Por outro lado, uma temperatura baixa, como 0, torna o modelo mais previsível e focado, resultando em respostas mais consistentes e menos aleatórias, sendo essencial para garantir a objetividade do modelo, o que é particularmente importante ao usar LLMs em tarefas objetivas de NLP, como a EL.

3.2.5 Atribuição de Entidades ESCO pelo LLM

Uma vez recebido o dicionário com as frases completas e as respetivas habilidades (coluna "*sentence*" e coluna "*skill*" do *dataset Synthetic-ESCO-skill-sentences*), é necessário extrair do *dataset Use-ESCO-data* as descrições oficiais ESCO de cada uma das habilidades identificadas pela base de dados FAISS. O objetivo desta metodologia é fornecer ao LLM informações suplementares necessárias para a tomada de decisão quanto à habilidade ESCO a ser atribuída.

O modelo pré-definido (*template*) de *prompt* escolhido será preenchido com a *Skill* de entrada a ser atribuída (*extracted_skill*); as habilidades entre as quais o LLM terá de escolher (*probable_entities*); as descrições oficiais de cada habilidade ESCO (*esco_skill_-description_text*); e os exemplos recebidos da base de dados (*esco_examples*).

3.3 Restrições

As principais restrições identificadas no desenvolvimento deste projeto foram relacionadas ao uso de modelos de LLMs menos avançados, que possuem um número menor de parâmetros treináveis. A versão utilizada do modelo *Llama2* conta com 7 mil milhões de parâmetros, sendo a menor versão disponível, enquanto há versões mais robustas com até 70 mil milhões de parâmetros do mesmo modelo. No entanto, essas versões mais avançadas requerem recursos de máquina significativamente maiores, como maior quantidade de RAM e melhor GPU, resultando em melhor desempenho, eficácia e velocidade de inferência. Devido às limitações de recursos disponíveis, foi necessário também utilizar mecanismos de quantização no modelo LLM, o que acarreta em algumas perdas de qualidade, tanto no modelo quanto no processo de inferência.

As limitações de recursos computacionais também impactaram o número de *tokens* que podiam ser processados pelo modelo LLM durante a inferência. O aumento da quantidade de *tokens* que podem ser passados no *contexto* via *prompt* foi um fator limitante

em abordagens que, por exemplo, incluiriam a frase completa onde a *Skill* foi extraída do texto não estruturado se encontrava, o que teria fornecido um contexto mais rico para o modelo identificar a *Skill* mais apropriada na tarefa de EL. Outra abordagem que poderia ser explorada, mas que foi restringida pelas limitações de entrada de *tokens*, seria a inclusão de *Few-Shot Learning* (FSL) na *prompt*. Essa técnica permitiria passar exemplos contextuais ao modelo, demonstrando o processo de atribuição de *Skills* na taxonomia ESCO. O estudo realizado sobre o estado da arte dos LLMs indicam que o uso de *few-shot learning* melhora a capacidade de raciocínio do modelo e o desempenho em tarefas de NLP, mesmo em casos onde o modelo não foi explicitamente treinado com os dados de entrada durante a fase de pré-treino, o que poderia ser uma abordagem promissora para aumento da performance dos LLMs em tarefas de *Entity Linking* (EL).

Capítulo 4

Avaliação do Sistema SA4S-RH

Neste capítulo será realizada a avaliação do sistema SA4S-RH, onde se avaliará a *pipeline* desenvolvida no projeto. Para tal, foi necessário desenvolver um *dataset* que permitisse a comparação entre os valores esperados das *skill* da taxonomia ESCO com o valor atribuído pela *pipeline*. Esse *dataset*, criado no âmbito deste projeto, foi concebido para testar a técnica de EL, associando uma entidade de *skill* extraída ao respetivo *skill* correto da taxonomia ESCO, servindo assim como *ground truth* para a avaliação.

4.1 Criação do dataset de avaliação

Foi criado o *dataset* de avaliação utilizando o *dataset Synthetic-ESCO-skill-sentences* (Decorte et al., 2023), também empregue no desenvolvimento da *pipeline* SA4S-RH. Esse *dataset* contém 10 frases de geradas sinteticamente para quase todas (99.5%) as *Skills* presentes na versão 1.1.0 da taxonomia ESCO, em inglês.

A criação do *dataset* de avaliação envolveu a utilização das três fases da *pipeline* SA4S-RH, sendo a primeira fase usada para extrair as *Skills* do texto não estruturado da coluna "*sentence*" do *dataset Synthetic-ESCO-skill-sentences* com a utilização do modelo de NER, *jjzha/escoxlmr_skill_extraction* desenvolvido por Zhang. Para cada *Skill* extraída do texto foi então criada uma nova entrada do *dataset* de avaliação correspondente à *Skill* extraída e a respetiva "*Skill*" da taxonomia ESCO da coluna "*skill*" do *dataset Synthetic-ESCO-skill-sentences*. As *Skills* extraídas da coluna "*sentence*" têm correspondência direta com as *Skills* da taxonomia ESCO da coluna "*skill*", uma vez que as frases foram geradas sinteticamente a partir das descrições oficiais das *Skills* da ESCO, o que permite considerar que as *Skills* extraídas possuem correspondência precisa com as *Skills* definidas na taxonomia, facilitando a validação e a análise dos resultados.

O *dataset* resultante foi filtrado para remover todos os elementos duplicados (com inclusão do original), tendo sido apenas consideradas válidas as entradas em que a coluna "*extracted_skill*" continha três ou mais palavras. Essa restrição foi necessária para reduzir

a entropia envolvida na associação de uma *Skill* extraída a uma entidade da taxonomia ESCO, uma vez que *Skills* com menos de 3 palavras não forneciam contexto suficiente para aferir a *Skill* ESCO de forma correta, garantindo assim maior precisão no processo de correspondência.

Este processo foi essencial para criar um *dataset* de verdade, contendo pares *Skill* extraida ("*extracted_skill*)" e a entidade *Skill* ESCO correta associada ("*esco_skill*"), que será populado com a utilização da pipeline.

O processo seguinte envolveu a avaliação da *pipeline* SA4S-RH através da utilização do *dataset* de avaliação, onde foram adicionadas as seguintes colunas:

- Coluna "*llm_el_esco*", Contendo a entidade *Skill* ESCO atribuída pela *pipeline*.
- Coluna "*vector_search_probable_esco_entities*", com a lista de *Skills* ESCO prováveis recebidas da pesquisa vetorial e ordenadas de forma descendente de pontuação (*score*) atribuído pela base de dados vetorial FAISS.
- Coluna "*esco_skill_in_vector_search*", onde será atribuído o valor booleano da correspondência da *Skill* ESCO escolhida pela *pipeline* SA4S-RH com a *Skill* ESCO correta da coluna "*esco_skill*".
- Coluna "*esco_skill_in_vector_search_in_position*", onde será atribuído o valor numero relativo à posição *N* onde se encontra a *Skill* ESCO correta na lista de *Skills* ESCO prováveis da coluna "*vector_search_probable_esco_entities*", no caso de acerto. Em caso de falha na atribuição pela *pipeline* a coluna è preenchida com o valor nulo (*NaN*).

Assim o *dataset* de avaliação foi populado com os resultados provenientes da segunda e terceira fases da *pipeline* SA4S-RH.

Na Tabela 4.1 é apresentada a tabela com exemplos do *dataset* de avaliação criado a partir do *dataset Synthetic-ESCO-skill-sentences*.

4.2 Performance da Pipeline SA4S-RH

Através da análise do *dataset* de avaliação populado com a *pipeline* SA4S-RH, foi possível extrair conclusões significativas sobre o desempenho da segunda e terceira fases da pipeline.

4.2.1 Avaliação da tarefa de filtro de entidades ESCO

Para avaliar a performance da segunda fase da *pipeline* SA4S-RH, correspondente à retribuição das entidade *Skills* da taxonomia ESCO prováveis, foi necessário comparar os valores de verdade (considerados corretos) das *Skills* ESCO com a lista de *Skills* ESCO

Extracted_skill	Esco_skill	Llm_el_esco	Vector_search_probable_esco_entities	Esco_skill_in_vector_search	Esco_skill_in_vector_search_position
reading and analyzing hallmarks on jewelry and other metal items.	read hallmarks	read hallmarks	['read hallmarks']	True	1
interpret and analyze food safety violations	evaluate retail food inspection findings	evaluate retail food inspection findings	['evaluate retail food inspection findings', 'assess food samples', 'food hygiene rules', 'assist in the development of standard operating procedures in the food chain']	True	1
properly align rubber plies	prepare rubber plies	prepare rubber plies	['build up rubber plies', 'brush rubber cement', 'prepare rubber plies', 'fasten rubber goods']	True	3

Tabela 4.1: Exemplos do *dataset* de avaliação

prováveis devolvida pela pesquisa vetorial, sendo esperado que no conjunto das entidades filtradas esteja contida a entidade correta. Assim, podemos observar que se a lista de entidades prováveis na coluna "*vector_search_probable_esco_entities*" contiver a entidade correta na coluna "*esco_skill*", este estará a funcionar corretamente, o que não limitará a decisão por parte do LLM.

Os resultados obtidos na analise de 2734 registos do *dataset* de avaliação mostraram que **em todos**, sem exceção, a *Skill* ESCO correta da coluna "*esco_skill*" está contida na lista de *Skills* prováveis da coluna "*vector_search_probable_esco_entities*", o que demonstra a excelente performance da utilização de pesquisa vetorial.

A figura 4.1 apresenta o gráfico de barras comparativo da posição *N* (*rank*) atribuída pela pesquisa vetorial à *Skill* ESCO correta, considerando apenas registos onde a *Skill* correta pertence ao conjunto de entidades *Skill* ESCO prováveis, tendo por isso sido considerados todos os registos.

Foi possível verificar que, na maioria dos casos (cerca de 80%), a *Skill* ESCO correta encontra-se na primeira posição entre as *Skills* ESCO mais prováveis (aquelas com maior *score*). Isso demonstra a eficácia da utilização da base de dados vetorial no processo de filtragem das entidades ESCO, atribuindo, na grande maioria, a maior probabilidade às entidades corretas.

Foi necessário realizar uma normalização dos dados, onde cada barra do gráfico corresponde à posição *N*-ésima das *Skills* corretas na taxonomia ESCO dentro da lista de possibilidades, em relação à coluna "*esco_skill*" (de *ground truth*). Assim a análise focou apenas nas entradas do conjunto de dados em que a coluna "*vector_search_probable_esco_entities*" contém *N* ou mais valores retornados de *Skills* prováveis da ESCO. Este

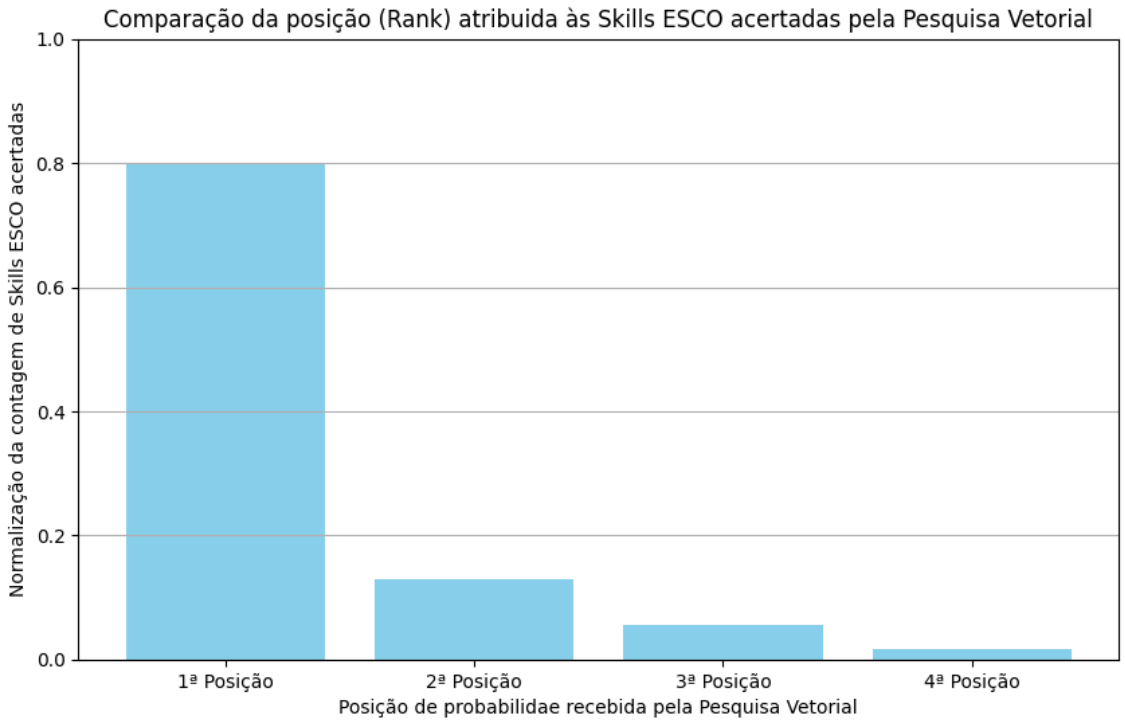


Figura 4.1: Gráfico de barras de comparação da posição (*Rank*) atribuída por pesquisa vetorial às *Skills ESCO* acertadas.

critério assegura que apenas as entradas do *dataset* com a quantidade igual ou superior de entidades na lista de ESCO prováveis, sejam consideradas na avaliação, proporcionando uma análise mais robusta e representativa do desempenho do filtro de entidades por parte da pesquisa vetorial.

4.2.2 Avaliação do LLM na Atribuição de Entidades ESCO

Para avaliar o desempenho do LLM *Llama2-7B* no processo de atribuição de entidades dentro do conjunto filtrado de *Skills* na taxonomia ESCO, foi realizada uma comparação entre os dados do *dataset* referentes às *Skills* ESCO de verdade (consideradas como corretas) e as correspondentes previsões de *Skills* da ESCO atribuídas pelo LLM no final da *pipeline*. A análise focou na taxa de acerto entre as competências atribuídas pelo LLM e as competências verdadeiras (*ground truth*) da ESCO presentes no *dataset*.

Foi necessário garantir a consistência dos dados para garantir a robustez da avaliação. Neste sentido, a verificação da correspondência entre as *Skills* atribuídas pelo modelo e as *Skills* verdadeiras foi realizada, dividindo-se os registos entre correspondências corretas e incorretas.

Os resultados obtidos foram os seguintes:

- Número total de registos onde o valor da coluna "*llm_el_esco*", atribuído pelo LLM, é valido: 2734

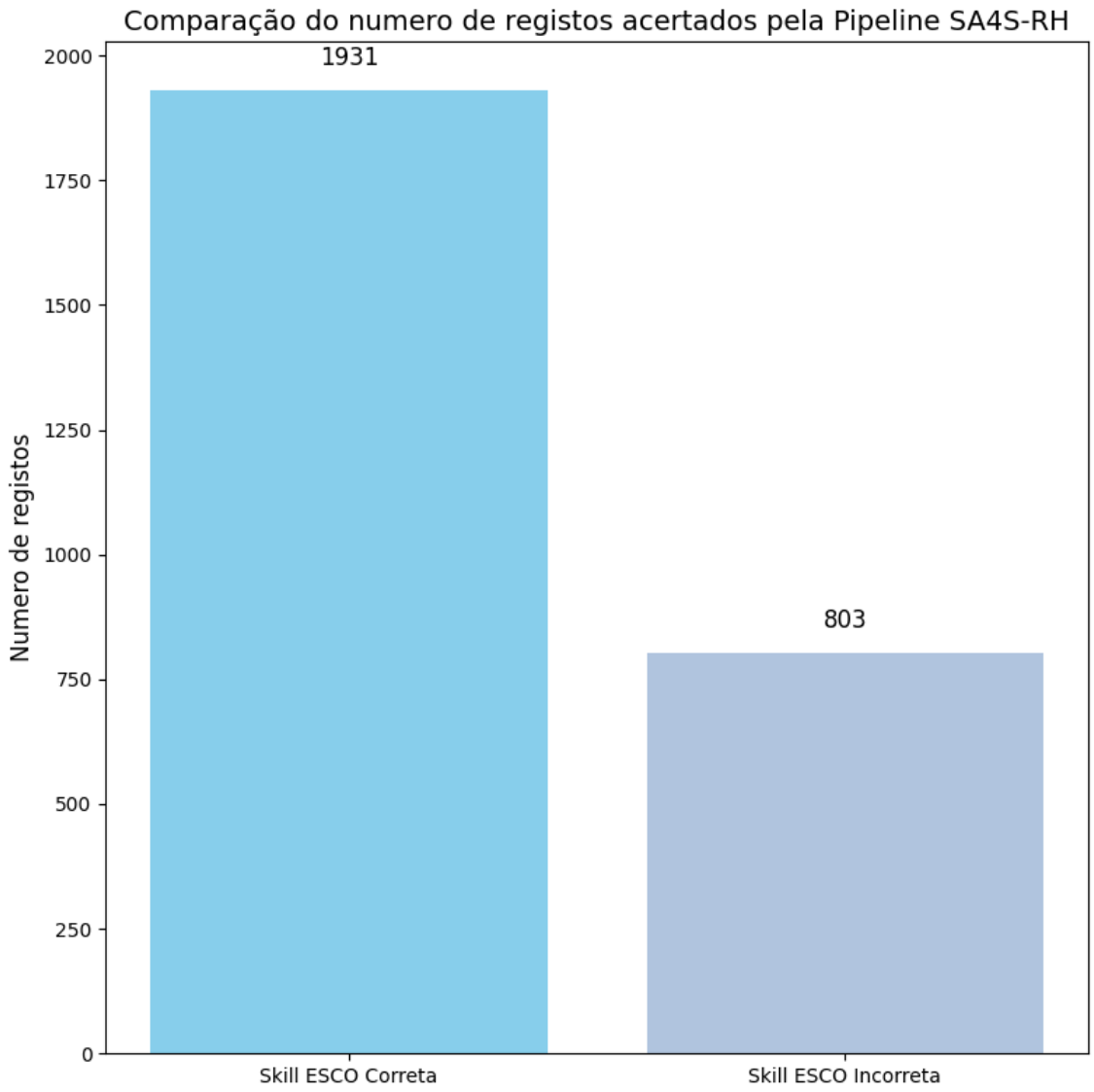


Figura 4.2: Comparação entre registos atribuídos com *Skills* ESCO corretas e incorretas pelo LLM Llama2-7B.

- Número de registos onde o valor da coluna de *ground truth* "*esco_skill*" **corresponde** ao valor da coluna "*llm_el_esco*": 1931
- Número de registos onde o valor da coluna de *ground truth* "*esco_skill*" **não corresponde** ao valor da coluna "*llm_el_esco*": 803

O gráfico de barras da figura 4.2 ilustra a comparação entre o número de registos corretamente atribuídos pelo LLM à entidade ESCO correta e os registos em que a atribuição foi incorreta. Esta visualização permite uma análise clara do desempenho do modelo, destacando a sua eficácia em tarefas de ligação de entidades.

É possível concluir que o LLM *Llama2-7B* demonstrou um desempenho consistente na tarefa de atribuição de entidades ESCO, com uma taxa de acerto significativa. Estes resultados reforçam o potencial do modelo na automação de processos de identificação

de competências no contexto do mercado de trabalho, contribuindo para uma análise mais eficiente das necessidades de competências a nível organizacional e industrial.

A partir dos dados analisados, a taxa de acerto do LLM *Llama2-7B* na atribuição de 2734 entidades ESCO foi de 70,63%, refletindo a boa precisão na identificação correta das *Skills*. Este desempenho positivo demonstra a eficácia do modelo em realizar atribuições de competências no contexto da taxonomia ESCO, sendo um indicador relevante para sua aplicação em cenários práticos.

O tempo médio total de processamento de cada inferência na *pipeline* foi de 9,29 segundos, necessário para processar uma entrada do *dataset* de avaliação.

No gráfico da figura 4.3, é possível observar a relação entre a precisão na atribuição de *Skills* tidas como corretas e a posição (*rank*) atribuída pela pesquisa vetorial FAISS, isto é, a taxa de acerto da *pipeline* em função do *rank* da *Skill* correta dentro da lista de entidades prováveis.

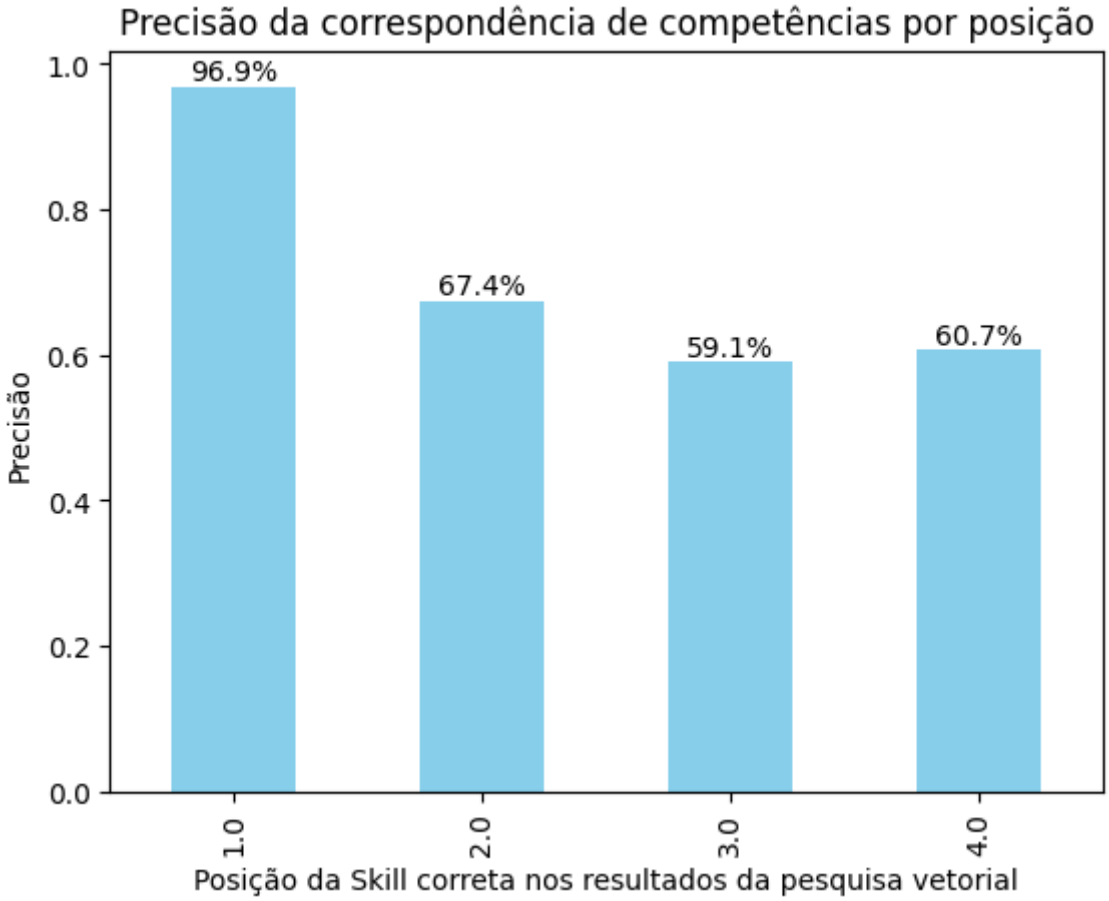


Figura 4.3: Comparação da precisão de correspondência da *pipeline* em função da posição (*Rank*) atribuída pela pesquisa vetorial com FAISS.

No gráfico, observa-se que quando a *Skill* considerada como correta se encontra na primeira posição (*rank*) na lista de entidades prováveis, a taxa de acerto da *pipeline* atinge 96,9%, evidenciando um excelente desempenho. Na segunda posição, a taxa de acerto

diminui para 67,4%, uma redução de cerca de 30%. Nas terceira e quarta posições, a precisão mantém-se quase uniforme, com valores de 59,1% e 60,7%, respetivamente. Esses resultados indicam que a taxa de acerto da *pipeline* é significativamente maior quando a *Skill* correta ocupa a primeira posições no *rank*, reduzindo-se gradualmente nas posições subsequentes. A uniformidade observada entre as terceira e quarta posições, com uma taxa de acerto próxima de 60%, sugere que a *pipeline* apresenta um bom desempenho principalmente quando a *Skill* correta está entre as duas primeiras posições do *rank* de probabilidades devolvido pela base de dados vetorial.

Para a criação dos gráficos, foi utiliza a biblioteca *Matplotlib*, amplamente utilizada para visualização de dados em *Python*, o que permitiu criar representações gráficas detalhadas e precisas do desempenho da *pipeline*.

4.3 Limitações na avaliação da pipeline SA4S-RH

A avaliação da tarefa de NER com o modelo *jjzha/escoxlmr_skill_extraction* de Zhang apresentou diversas limitações que dificultaram a validação precisa e confiável dos resultados na extração de entidades *Skill*. A ausência de um conjunto de dados manualmente anotado, abrangente e específico para as necessidades deste projeto, complicou a validação rigorosa das *Skills* extraídas. Esse tipo de anotação manual seria essencial para assegurar que as *Skills* extraídas correspondem diretamente às *Skills* da taxonomia ESCO, um requisito crucial para garantir uma comparação justa e precisa entre as predições da *pipeline* SA4S-RH e as referências verdadeiras. No entanto, devido à complexidade e ao esforço envolvido na criação de um conjunto de dados anotado manualmente, não foi possível implementá-lo neste contexto.

Para mitigar essa limitação, foi utilizado o *dataset Synthetic-ESCO-skill-sentences*, onde se pode sobrepujar que as *Skills* extraídas a partir da coluna "*sentence*" têm uma correspondência direta com as *Skills* da taxonomia ESCO da coluna "*skill*", dado que essas frases foram geradas sinteticamente a partir das descrições oficiais das *Skills* ESCO. Embora este *dataset* tenha permitido avaliar o desempenho do modelo, a sua natureza sintética representa uma limitação significativa. O uso de dados sintéticos, apesar de útil, não captura totalmente a complexidade e a variabilidade de dados reais encontrados em descrições de vagas de emprego ou em CVs, que seriam ideais para esta tarefa. A utilização de dados reais permitiria uma avaliação mais robusta e realista da capacidade e eficácia do modelo *jjzha/escoxlmr_skill_extraction* para extrair *Skills* relevantes e da *pipeline* SA4S-RH na tarefa de EL de atribuição de entidades ESCO, no contexto do mercado de trabalho. Esta limitação, portanto, restringe a generalização dos resultados obtidos, já que a aplicação em cenários reais pode apresentar desafios adicionais que não foram considerados na avaliação com dados sintéticos.

Além disso, foi inviável realizar uma avaliação completa da eficácia de diferentes es-

tratégias de *prompting* com o LLM. Não existiam convenções suficientemente robustas que permitissem uma análise justa e padronizada das respostas geradas pelo modelo a partir de diferentes *prompts*. A ausência de tais convenções dificultou a comparação objetiva do texto de saída do LLM no final da inferência.

Outro desafio significativo foi a impossibilidade de testar configurações de *prompting*, como *Few-Shot Learning* (FSL), onde o modelo receberia exemplos práticos para abordar as tarefas. Esta limitação ocorreu devido à restrição de memória (limite de *tokens* de *input* passados ao contexto) e de RAM na GPU, que impossibilitaram a criação de *prompts* mais elaboradas.

Também é relevante notar que qualquer comparação entre diferentes configurações de *prompting*, modelos LLM, ou ajustes específicos do modelo LLM teria exigido uma reavaliação completa utilizando todo o *dataset*. No entanto, isso mostrou-se inviável devido ao tempo necessário para realizar cada inferência, sendo em média 9,29 segundos. A execução de múltiplas iterações sobre o *dataset* de avaliação para cada atribuição de entidade pela taxonomia ESCO foi limitada pelos recursos computacionais disponíveis.

Adicionalmente, a limitação dos recursos de *hardware* também foi um obstáculo importante. Não foi possível processar todas as 138260 entradas do *dataset* de treino devido às restrições computacionais, tanto em termos do tempo disponível nas instâncias de GPU no *Google Colab*, quanto pela velocidade de inferência necessária para a atribuição de cada entidade. Estes fatores, combinados com o tempo elevado de processamento por inferência, limitaram o alcance e a exaustividade dos testes, impondo restrições ao número de exemplos que poderiam ser processados em tempo útil.

Capítulo 5

Conclusão

Neste capítulo são apresentadas as conclusões obtidas com o desenvolvimento do sistema *Sistema Assistido para Seleção de Recursos Humanos* (SA4S-RH), discutindo os resultados obtidos, as limitações e propondo possíveis melhorias em trabalhos futuros. O capítulo permite a reflexão sobre a utilização de LLMs em tarefas de EL, especificamente aplicadas ao domínio do mercado de trabalho.

5.1 Síntese de Resultados

Os resultados deste trabalho demonstraram um potencial promissor na utilização de LLMs em tarefas de EL. Com o uso de um *dataset* sintético, foi alcançada uma taxa de acerto de 70,63%, o que sugere a viabilidade da *pipeline* desenvolvida para atribuição de entidades da taxonomia ESCO. Embora a precisão não seja perfeita, esse índice é um indicativo positivo de que o uso de LLMs pode contribuir significativamente para o aprimoramento dos processos de EL no contexto do mercado de trabalho.

Além disso, este trabalho revelou-se promissor na aplicação de metodologias baseadas em pesquisa vetorial para uma filtragem confiável de entidades numa KB, como a ESCO. Utilizando a distância euclidiana entre os vetores de *embedding*, foi possível realizar uma redução eficaz do conjunto de entidades prováveis a serem atribuídas na tarefa de EL. Foi comprovado que, em todos os casos de teste, esta abordagem incluiu a entidade ESCO correta entre as entidades sugeridas como mais prováveis. Foi também possível verificar que o desempenho da base de dados vetorial em filtrar e atribuir *ranks* às entidades prováveis afetou de forma direta a performance do LLM no processo de atribuição de entidades.

Esta abordagem mostrou-se eficaz em mitigar as limitações dos LLMs em tarefas de EL com um elevado número de entidades, revelando-se uma estratégia promissora para a utilização de LLMs. A aplicação desta metodologia contribuiu significativamente para melhorar a precisão na atribuição de *skills*, reduzindo a complexidade da tarefa e tornando-a mais adaptável a diferentes cenários.

5.2 Limitações e Trabalho Futuro

É importante reconhecer as limitações deste estudo, como o uso de dados sintéticos, que não refletem na totalmente a diversidade e nuances dos dados reais, como descrições de vagas de emprego ou CVs. Seria importante a utilização de modelos de LLM mais avançados e otimizados para a velocidade de inferência, ideais para a implementação da metodologia num sistema de ATS real.

Seria igualmente interessante explorar o desempenho do modelo em casos de *fine-tuning*, especificamente aplicando técnicas de *instruction tuning*. Nesse cenário, pares de *input* poderiam ser criados utilizando a mesma *prompt* desenvolvida para o sistema *Sistema Assistido para Seleção de Recursos Humanos* (SA4S-RH), onde o modelo teria como resposta a atribuição correta de uma *Skill* à entidade da taxonomia ESCO correspondente. Essa abordagem poderia aumentar a precisão do modelo ao adaptá-lo mais profundamente à tarefa específica de EL no contexto do mercado de trabalho.

Melhorias futuras podem incluir a utilização de dados reais anotados para treinar e avaliar o modelo, substituindo os dados sintéticos utilizados no presente trabalho. Isso proporcionaria uma validação mais robusta e realista do desempenho do sistema. Além disso, técnicas de *prompting*, como o *few-shot learning*, podem ser exploradas para reduzir a necessidade de grandes volumes de dados de treino, tornando o modelo mais eficiente e adaptável às variações linguísticas e contextuais observadas nos textos reais, como descrições de vagas de emprego e CVs. Essas melhorias poderiam incrementar ainda mais a precisão e a utilidade prática do sistema SA4S-RH, otimizando ainda mais processos de seleção e recrutamento.

Para a avaliação do *pipeline*, será importante também comparar com outros modelos de NLP desenvolvidos para a tarefa de EL, em particular os modelos baseados em *GENRE* e *BLINK*, propostos por Zhang et al., que foram treinados para a taxonomia ESCO. Estes modelos, no entanto, não foram abrangidos neste trabalho devido a limitações de *hardware* que impediram a realização do processo de treino e avaliação dos mesmos.

Referências

Josette Bettany-Saltikov. Learning how to undertake a systematic review: part 1. *Nursing standard (Royal College of Nursing (Great Britain) : 1987)*, 24:47–55; quiz 56, 08 2010. doi: 10.7748/ns2010.08.24.50.47.c7939. 23

Nicola De Cao, Gautier Izacard, Sebastian Riedel, and Fabio Petroni. Autoregressive entity retrieval, 2021. URL https://arxiv.org/abs/2010.00904. 21

Yihan Cao, Xu Chen, Lun Du, Hao Chen, Qiang Fu, Shi Han, Yushu Du, Yanbin Kang, Guangming Lu, and Zi Li. Tarot: A hierarchical framework with multitask co-pretraining on semi-structured data towards effective person-job fit, 2024. URL https://arxiv.org/abs/2401.07525. 14, 15, 18, 19

Ricardo Manuel da Silva Ferreira, Michael Canesche, and Jeronimo Costa Penha. Google colab para ensino de computação, 2023. 37

Jens-Joris Decorte, Severine Verlinden, Jeroen Van Hautte, Johannes Deleu, Chris Develder, and Thomas Demeester. Extreme multi-label skill extraction training using large language models, 2023. URL https://arxiv.org/abs/2307.10778. 16, 20, 23, 40, 51

Yifan Ding, Qingkai Zeng, and Tim Weninger. Chatel: Entity linking with chatbots, 2024. URL https://arxiv.org/abs/2402.14858. 14, 17, 18, 21, 43

Cicero Nogueira dos Santos and Victor Guimarães. Boosting named entity recognition with neural character embeddings, 2015. URL https://arxiv.org/abs/1505.05008. 12

Yingpeng Du, Di Luo, Rui Yan, Hongzhi Liu, Yang Song, Hengshu Zhu, and Jie Zhang. Enhancing job recommendation through llm-based generative adversarial networks, 2023. URL https://arxiv.org/abs/2307.10747. 16, 17, 20, 21

Chengguang Gan, Qinghao Zhang, and Tatsunori Mori. Application of llm agents in recruitment: A novel framework for resume screening, 2024. URL https://arxiv.org/abs/2401.08315. 8, 9, 15, 16, 17, 19, 20

Jiafeng Guo, Yixing Fan, Qingyao Ai, and W. Bruce Croft. A deep relevance matching model for ad-hoc retrieval. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, CIKM'16. ACM, October 2016. doi: 10.1145/2983323.2983769. URL http://dx.doi.org/10.1145/2983323.2983769. 12

Jeff Johnson, Matthijs Douze, and Hervé Jégou. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547, 2019. 39

Nam Ho Koh, Joseph Plata, and Joyce Chai. Bad: Bias detection for large language models in the context of candidate screening, 2023. URL https://arxiv.org/abs/2305.10407. 15, 17, 19, 22

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners, 2023. URL https://arxiv.org/abs/2205.11916. 10, 11

Belinda Z. Li, Sewon Min, Srinivasan Iyer, Yashar Mehdad, and Wen-tau Yih. Efficient one-pass end-to-end entity linking for questions. In *EMNLP*, 2020. 21

Xuezhe Ma and Eduard Hovy. End-to-end sequence labeling via bi-directional lstm-cnns-crf, 2016. URL https://arxiv.org/abs/1603.01354. 12

Shervin Minaee, Tomas Mikolov, Narjes Nikzad, Meysam Chenaghlu, Richard Socher, Xavier Amatriain, and Jianfeng Gao. Large language models: A survey, 2024. URL https://arxiv.org/abs/2402.06196. 7, 8, 9, 10, 11

Daniel W. Otter, Julian R. Medina, and Jugal K. Kalita. A survey of the usages of deep learning in natural language processing, 2019. URL https://arxiv.org/abs/1807.10854. 5, 6, 7, 8, 11, 12, 13

Matthew J. Page, Joanne E. McKenzie, Patrick M. Bossuyt, Isabelle Boutron, Tammy C. Hoffmann, Cynthia D. Mulrow, Larissa Shamseer, Jennifer M. Tetzlaff, Elie A. Akl, Sue E. Brennan, Roger Chou, Julie Glanville, Jeremy M. Grimshaw, Asbjørn Hróbjartsson, Manoj M. Lalu, Tianjing Li, Elizabeth W. Loder, Evan Mayo-Wilson, Steve McDonald, Luke A. McGuinness, Lesley A. Stewart, James Thomas, Andrea C. Tricco, Vivian A. Welch, Penny Whiting, and David Moher. The prisma 2020 statement: an updated guideline for reporting systematic reviews. *Systematic Reviews*, 10(1):89, Mar 2021. ISSN 2046-4053. doi: 10.1186/s13643-021-01626-4. URL https://doi.org/10.1186/s13643-021-01626-4. 25

Jane Phillips and Chet Robie. Can a computer outfake a human? *Personality and Individual Differences*, 217:112434, 2024. ISSN 0191-8869. doi: https://doi.org/10.1016/

j.paid.2023.112434. URL https://www.sciencedirect.com/science/article/pii/S0191886923003574. 17, 19

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019. URL https://arxiv.org/abs/1908.10084. 39

Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition, 2003. URL https://arxiv.org/abs/cs/0306050. 14

Liqun Shan, Yanchang Liu, Min Tang, Ming Yang, and Xueyuan Bai. Cnn-bilstm hybrid neural networks with attention mechanism for well log prediction. *Journal of Petroleum Science and Engineering*, 205:108838, 2021. ISSN 0920-4105. doi: https://doi.org/10.1016/j.petrol.2021.108838. URL https://www.sciencedirect.com/science/article/pii/S092041052100499X. 12

Panagiotis Skondras, Panagiotis Zervas, and Giannis Tzimas. Generating synthetic resume data with large language models for enhanced job description classification. *Future Internet*, 15(11), 2023. ISSN 1999-5903. doi: 10.3390/fi15110363. URL https://www.mdpi.com/1999-5903/15/11/363. 8, 15, 16, 17, 18, 19, 22

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023. URL https://arxiv.org/abs/2307.09288. 31, 40

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu,

Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.emnlp-demos.6. 8, 9, 38

Likang Wu, Zhaopeng Qiu, Zhi Zheng, Hengshu Zhu, and Enhong Chen. Exploring large language model for graph data understanding in online job recommendations, 2023. URL https://arxiv.org/abs/2307.05722. 8, 9, 15, 16, 17

Hamed Zamani, Mostafa Dehghani, W. Bruce Croft, Erik Learned-Miller, and Jaap Kamps. From neural re-ranking to neural ranking: Learning a sparse representation for inverted indexing. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, CIKM '18, page 497–506, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450360142. doi: 10.1145/3269206.3271800. URL https://doi.org/10.1145/3269206.3271800. 12

Mike Zhang. Computational job market analysis with natural language processing, 2024. URL https://arxiv.org/abs/2404.18977. 1, 13, 14, 21, 30, 40, 45, 51, 57

Mike Zhang, Rob van der Goot, and Barbara Plank. Escoxlm-r: Multilingual taxonomy-driven pre-training for the job market domain, 2023. URL https://arxiv.org/abs/2305.12092. 45

Mike Zhang, Rob van der Goot, and Barbara Plank. Entity linking in the job market domain. In Yvette Graham and Matthew Purver, editors, *Findings of the Association for Computational Linguistics: EACL 2024*, pages 410–419, St. Julian's, Malta, March 2024a. Association for Computational Linguistics. URL https://aclanthology.org/2024.findings-eacl.28. 13, 14, 21, 22, 60

Zhen Zhang, Yuhua Zhao, Hang Gao, and Mengting Hu. Linkner: Linking local named entity recognition models to large language models using uncertainty, 2024b. URL https://arxiv.org/abs/2402.10573. 43

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, Yifan Du, Chen Yang, Yushuo Chen, Zhipeng Chen, Jinhao Jiang, Ruiyang Ren, Yifan Li, Xinyu Tang, Zikang Liu, Peiyu Liu, Jian-Yun Nie, and Ji-Rong Wen. A survey of large language models, 2024. 7, 8, 9, 10

Zhi Zheng, Zhaopeng Qiu, Xiao Hu, Likang Wu, Hengshu Zhu, and Hui Xiong. Generative job recommendations with large language model, 2023. URL https://arxiv.org/abs/2307.02157. 8, 9, 15, 17, 19, 20