Avaliação de um Sistema RAG Aplicado ao Domínio Jurídico Brasileiro

Gabriel Inagaki Marcelino

[1]Departamento de Ciências da Computação e Estatística – Universidade Estadual Paulista (UNESP)
Caixa Postal 15054-000 – São José do Rio Preto – SP – Brazil

gabriel.inagaki@unesp.br

Abstract. *This work presents the development and evaluation of a Retrieval-Augmented Generation (RAG) system applied to the Brazilian legal domain, represented by a reduced legislative corpus and powered by a Llama-family model fine-tuned through LoRA. The objective is to examine the extent to which a RAG system can generate adequate answers using retrieved context. The results showed limited performance: the model frequently ignored the provided context, struggled to identify relevant information, and at times produced generic outputs or informational hallucinations—even when the correct passage was included among the retrieved documents. This study contributes as practical documentation of the real challenges encountered when training and evaluating language models in the legal context, highlighting limitations and suggesting directions for future work.*

Resumo. *Este trabalho apresenta a construção e avaliação de um sistema de Retrieval Augmented Generation (RAG) aplicado ao domínio jurídico brasileiro, representado por um corpus legislativo reduzido, utilizando um modelo da família Llama ajustado via LoRA. O objetivo é investigar em que medida um sistema RAG é capaz de gerar respostas adequadas utilizando contexto recuperado. Os resultados mostraram desempenho limitado: o modelo frequentemente ignorou o contexto fornecido, apresentou dificuldades em localizar informações relevantes e, por vezes, produziu respostas genéricas ou alucinações informacionais, mesmo quando o trecho correto estava entre os recuperados. O estudo contribui como documentação prática dos desafios reais enfrentados ao treinar e avaliar modelos de língua em cenários jurídicos, ressaltando limitações e sugerindo direções para trabalhos futuros.*

1. Introdução

O acesso transparente à legislação é requisito fundamental para a consolidação do Estado de Direito e da cidadania. No contexto brasileiro, a quantia de normas em vigor, sua natureza dinâmica e a complexidade de sua redação tornam tanto a consulta quanto a interpretação desafiadoras (Moreira et al. 2011). Apesar de iniciativas estatais de acessibilidade por meios digitais, o formato em que essas informações são apresentadas frequentemente apresenta atritos para interpretação.

Paralelamente, avanços em Processamento de Linguagem Natural (PLN) e em Modelos de Linguagem de Grande Escala (LLMs) têm transformado a forma como documentos de diferentes naturezas podem ser explorados (Hassani and Silva 2023). O

emprego de tais tecnologias abre possibilidades de pesquisa automatizada, resumos e explicações de documentos legais e respostas a consultas específicas.

Neste trabalho, é apresentada uma metodologia para a construção de um LLM especializado no domínio jurídico nacional, desde a contrução do corpus jurídico, estruturado a partir de textos legislativos brasileiros, até o treinamento do LLM utilizando tal base.

Este estudo busca oferecer um recurso de dados reprodutível e adaptável, mesmo que relativamente pequeno. Ao mesmo tempo, busca-se fomentar a intersecção entre as áreas de Ciência de Dados, Inteligência Artificial e Direito, promovendo meios de facilitar o acesso popular às leis vigentes.

2. Objetivos Gerais e Específicos

Este trabalho tem como objetivo principal avaliar o desempenho de um modelo de linguagem de grande escala (LLM) aplicado a um corpus jurídico brasileiro de pequeno porte, construído a partir de textos legislativos oficiais. Busca-se verificar em que medida tal modelo ajustado é capaz de compreender, recuperar e responder a consultas sobre o conteúdo. Mais especificamente, o estudo pretende:

- Desenvolver uma metodologia de preparação de dados jurídicos, incluindo a extração e estruturação de leis brasileiras em formato legível por modelos de linguagem;
- Gerar um conjunto de pares de instrução e resposta (baseado no formato *instruction tuning*), representativo de questões e explicações de artigos de lei;
- Realizar o ajuste fino (*fine-tuning*) de um modelo LLaMA (Touvron et al. 2023) sobre o corpus gerado, avaliando sua performance em inferências de caráter legislativo;
- Explorar a integração de mecanismos de recuperação de informação (*Retrieval-Augmented Generation* – RAG) para complementar o modelo com acesso direto ao texto legal durante a inferência;
- Analisar as respostas do modelo, identificando limitações e potenciais para estudos futuros.

3. Fundamentação Teórica

3.1. Modelos de Linguagem e Fine-tuning

Modelos de linguagem de grande escala (LLMs — *Large Language Models*) são capazes de capturar padrões linguísticos complexos e realizar tarefas variadas, como sumarização, tradução e resposta a perguntas, sem necessidade de instruções explícitas (Brown et al. 2020). Sua aplicação no domínio jurídico apresenta um grande potencial, sobretudo em tarefas de extração e recuperação de informações, explicação de textos legais e apoio à pesquisa normativa.

LLMs como o LLaMA (*Large Language Model Meta AI*) são arquiteturas fundamentadas em Transformers (Vaswani et al. 2017), treinadas sobre grandes volumes de texto não supervisionado (Touvron et al. 2023). Apesar de seu vasto conhecimento

linguístico, sua aplicação em domínios especializados — como o jurídico — pode apresentar resultados mais adequados através de um processo de adaptação chamado *fine-tuning*. Esse processo consiste em ajustar os pesos do modelo a partir de um conjunto de dados anotado que representa o domínio de interesse.

3.2. Recuperação de Informação e o Modelo RAG

O paradigma de *Retrieval-Augmented Generation* (RAG) (Lewis et al. 2020) introduz uma abordagem que combina modelos generativos com mecanismos de busca em bases de conhecimento. Enquanto o LLM é responsável pela geração textual, o componente de recuperação busca informações relevantes em uma base de documentos para enriquecer a resposta. Assim, o modelo mantém sua função de gerador, mas ancorando a inferência em dados pré-estabelecidos.

No domínio jurídico, a aplicação do RAG se mostra particularmente promissora. As leis e normas são altamente interdependentes e constantemente sofrem alterações, o que dificulta o aprendizado puramente paramétrico. O uso de recuperação externa permite que o modelo consulte diretamente trechos normativos durante a inferência.

3.3. Base de Dados

A legislação brasileira é composta por milhares de normas e dispositivos interligados, dispersos em diferentes repositórios e formatos. Embora existam portais oficiais, como o *Planalto.gov.br*, esses recursos não são organizados de forma estruturada para o uso em modelos de PLN. A preparação de dados para aplicações em IA requer etapas de limpeza, normalização e segmentação textual, de modo a tornar o conteúdo acessível e indexável (Hassani and Silva 2023).

4. Material e método

A Figura 1 apresenta o fluxograma geral da metodologia desenvolvida, englobando coleta e processamento dos textos legislativos, indexação semântica FAISS, geração do corpus supervisionado com RAG e, por fim, o ajuste fino supervisionado (SFT) com LoRA.

4.1. Ambiente de desenvolvimento

Os experimentos foram conduzidos no computador Ada Lovelace do Laboratório de Inovação e Desenvolvimento em Inteligência Artificial do Ibilce (LIDIA), utilizando uma placa gráfica NVIDIA GeForce RTX 4090. O sistema operacional base utilizado foi Ubuntu 24.04.2 LTS com linguagem Python 3.12.12.

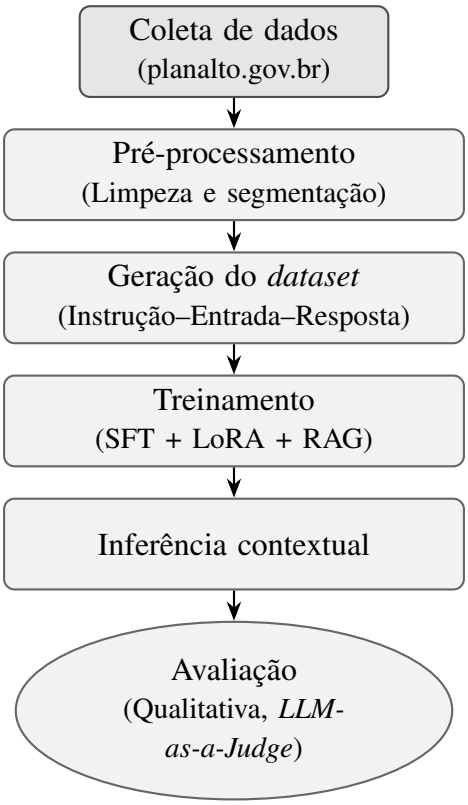
As especificações da máquina Ada Lovelace constam na Tabela 1:

Tabela 1. Especificações da máquina utilizada

	Processador	GPU
Computador LIDIA	Intel i9-13900K (32) @ 5.8GHz	RTX 4090 24GB (2×)

Fonte: Autoria própria

Figura 1. Fluxograma do processo metodológico.



Fonte: Autoria própria

4.2. Linguagem, bibliotecas e *frameworks*

O ambiente de desenvolvimento foi composto pelas seguintes bibliotecas e *frameworks*:

- `torch` (Paszke et al. 2019), versão 2.8.1;
- `transformers` (Wolf et al. 2020), versão 4.57.1;
- `trl` (von Werra et al. 2020), utilizada para o treinamento supervisionado (SFT), versão 0.23.0;
- `unsloth` (Daniel Han and team 2023), *framework* facilitador de treinamento de LLMs e aplicação de LoRA, versão 2025.11.1;
- `sentence-transformers` (Reimers and Gurevych 2019a), para geração de *embeddings* semânticos, versão 5.1.2;

Uma lista completa com todas as bibliotecas utilizadas e suas versões está disponível no repositório do trabalho.

4.3. Coleta e Processamento de Textos Legislativos

A base de dados utilizada neste trabalho foi construída a partir de fontes e textos oficiais. A base é constituída por leis ordinárias, decretos e artigos de códigos e estatutos populares. Devido à complexidade do processamento de todo o corpus legal brasileiro, foram selecionados apenas 15 textos legais populares, sendo eles: Código Civil, Código de Defesa do Consumidor, Código de Processo Civil, Código de Processo Penal, Código Penal,

Código Tributário Nacional, Consolidação das Leis do Trabalho (CLT), Constituição Federal de 1988, Estatuto da Criança e do Adolescente (ECA), Estatuto da Igualdade Racial, Estatuto do idoso, Lei de Acesso à Informação, Lei de Drogas, Lei Geral de Proteção de Dados (LGPD) e Lei Maria da Penha.

A etapa de coleta de dados consistiu na extração automatizada dos textos a partir do Portal da Legislação do Governo Federal. A extração foi feita com Python, utilizando as bibliotecas `requests` (Reitz et al. 2025) e `beautifulsoup` (Richardson 2025) para o download e processamento da estrutura HTML das páginas. Cada norma teve seu respectivo link extraído manualmente, considerando que a estrutura de URLs do portal não segue um padrão fixo.

4.4. Indexação Semântica com FAISS

Foi adotado o modelo `paraphrase-multilingual-mpnet-base-v2` (Reimers and Gurevych 2019b) para geração dos *embeddings* dos *chunks*. Os vetores foram normalizados e organizados em um índice FAISS (Douze et al. 2025) do tipo `IndexFlatIP`, adequado para buscas por similaridade coseno.

4.5. Geração do Dataset Supervisionado

Foi construído um *dataset* de ajuste fino supervisionado (SFT) baseado em RAG. O processo ocorre em três etapas principais:

1. **Geração de perguntas**: Para cada artigo, foram geradas três questões utilizando *templates* do tipo factual e interpretativo:
    - "O que diz o artigo *n* da *lei*?" (factual)
    - "Explique o conteúdo do artigo *n* da *lei*." (interpretativo)
    - "Quais direitos ou deveres o artigo *n* estabelece?" (interpretativo)
2. **Construção do contexto RAG**: O contexto final foi formado por:
    - o próprio artigo alvo,
    - artigos correlatos semanticamente dentro da mesma lei,
    - trechos recuperados via FAISS (desde que da mesma norma).

    O tamanho máximo do contexto foi limitado a 4500 caracteres, evitando exemplos excessivamente longos.
3. **Geração da resposta-alvo**: Para perguntas interpretativas, utilizou-se o próprio modelo Llama 3.1-8B executado localmente para produzir respostas sintéticas, utilizando de engenharia de *prompt* para adequar melhor as repostas ao uso.

O modelo treinado foi o **Meta-Llama-3.1-8B-Instruct**, o mesmo utilizado nesta Subseção para geração das respostas sintéticas.

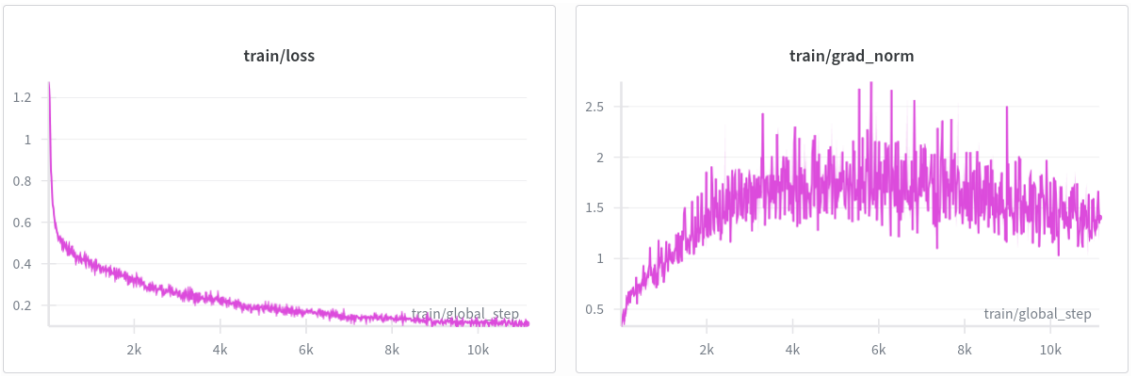
5. Experimentos e Resultados

Nesta seção são apresentados os resultados obtidos pela abordagem de Recuperação e Geração (RAG) aplicada ao corpus legislativo brasileiro. Os experimentos foram conduzidos utilizando o pipeline descrito na Seção 4.

5.1. Treinamento

O treinamento do modelo foi feito por 10 épocas, compostas por 11135 *steps*. O treinamento durou em torno de 10 horas, terminando com uma *loss* de treinamento de 0.22. Os gráficos com os valores da *loss* e norma do gradiente constam na Figura 2.

Figura 2. Gráficos produzidos com o treinamento. À esquerda a evolução da *loss* e à direita a evolução da norma do gradiente



Fonte: Autoria própria

5.2. *LLM-as-a-Judge*

Seguindo o paradigma *LLM-as-a-Judge*, foi utilizado o ChatGPT (GPT-5.1) (OpenAI 2024) para avaliar as respostas geradas. A seguir são apresentadas as perguntas feitas, o contexto, contendo as informações buscadas pelo mecanismo de recuperação dada a pergunta, a resposta dada pelo modelo treinado, e por fim a avaliação do *LLM-as-a-Judge* com pontuações variando de 0 até 10. As avaliações foram geradas seguindo os critérios correção factual, coerência, clareza e utilidade jurídica.

1. O que diz o artigo 1 do Código de Defesa do Consumidor?
    - **Contexto (truncado):** Fonte: CDC Art. 54-D (... deveres na oferta de crédito ...); Fonte: Código Civil Art. 1004 (... obrigações dos sócios ...)
    - **Resposta do modelo avaliado (truncada):** O artigo 1 do Código de Defesa do Consumidor não está disponível na fonte fornecida. No entanto, posso sugerir que o artigo 1 [...] estabelece que ”Consumidor é a pessoa física ou jurídica a quem é destinada a oferta de serviços ou produtos [...]”
    - **Avaliação *LLM-as-a-Judge*:** A resposta apresenta informações inventadas sobre o conteúdo do Art. 1º do CDC, confundindo-o com os artigos 2º e 3º. Embora o texto seja claro, falta correção factual e há pouca utilidade jurídica, visto que fornece definições erradas e não sustentadas pelo contexto, configurando alucinação.

    Correção factual: 2 **Coerência:** 4 **Clareza:** 6 **Utilidade jurídica:** 1
2. Sobre o que se trata o artigo 1746 do Código Civil?
    - **Contexto (truncado):** Fonte: CC Art. 1671 (... extinta a comunhão, divisão de ativo e passivo ...); Fonte: CLT Art. 215 (... revogado ...)
    - **Resposta do modelo avaliado (truncada):** O artigo 1746 do código civil não é mencionado na pergunta, mas o artigo 1671 está relacionado à comunhão de bens [...] Já o artigo 215 da CLT foi revogado [...] Verifique a seção correta do código civil para obter a informação solicitada. [...]
    - **Avaliação *LLM-as-a-Judge*:** A resposta ignora a pergunta principal e não tenta responder sobre o Art. 1746, limitando-se a repetir partes do contexto

irrelevante. Há erro lógico ("não é mencionado na pergunta" quando a pergunta menciona o artigo). A resposta não oferece conteúdo jurídico útil e contém divagações e sugestões genéricas.

Correção factual: 1 **Coerência:** 2 **Clareza:** 5 **Utilidade jurídica:** 1

Em nenhum dos experimentos conduzidos a recuperação RAG trouxe contexto relevante para a pergunta. Considerando que o modelo foi treinado apenas em situações onde o contexto enriquecia a resposta, foram feitos também experimentos dando manualmente um contexto adequado.

1. Explique o conteúdo do artigo 922 da Consolidação das Leis do Trabalho – CLT
   - **Contexto (truncado):** Fonte: CLT Art. 922 (... art. 301 regerá apenas relações de emprego iniciadas após a vigência da CLT; seguido de quadro setorial da indústria da alimentação ...);
   - **Resposta do modelo avaliado (truncada):** O artigo 922 da CLT não existe, mas trata sobre a responsabilidade do empregador por danos causados ao empregado [...] mesmo que decorrentes de terceiros [...] é um dispositivo importante para proteção dos trabalhadores [...]
   - **Avaliação *LLM-as-a-Judge*:** A resposta é gravemente incorreta: afirma que o artigo "não existe" e em seguida inventa um conteúdo totalmente alheio ao texto. O Art. 922 real está no contexto (...) Trata-se de alucinação jurídica completa, sem relação com o conteúdo recuperado.

   Correção factual: 0 **Coerência:** 2 **Clareza:** 4 **Utilidade jurídica:** 0
2. Explique o conteúdo do artigo 26 do Código Civil
   - **Contexto (truncado):** Fonte: CC Art. 26 (... após 1 ano da arrecadação dos bens do ausente, ou 3 anos se deixou representante, interessados podem requerer declaração de ausência e abertura provisória da sucessão ...);
   - **Resposta do modelo avaliado (truncada):** O artigo 26 do Código Civil não está presente na lista de artigos fornecida. [...]
   - **Avaliação *LLM-as-a-Judge*:** A resposta é totalmente desconectada do contexto, que contém exatamente o Art. 26 solicitado. O modelo afirma repetidamente que o artigo "não está presente", apesar de ele estar explícito na fonte (...) Trata-se de falha grave de leitura e compreensão do contexto, resultando em utilidade jurídica nula.

   Correção factual: 0 **Coerência:** 2 **Clareza:** 4 **Utilidade jurídica:** 0

5.3. Avaliação Qualitativa

Além da abordagem *LLM-as-a-Judge*, foi realizada uma análise manual das respostas geradas. A Tabela 2 apresenta alguns exemplos de resposta gerados pelo modelo.

Analisando as respostas geradas pelo modelo treinado na Tabela 2, foi observado que o modelo aprendeu com sucesso a incluir a busca no *retrieval* em sua resposta. Entretanto, apesar de procurar, em nenhum dos casos foi capaz de responder de forma satisfatória. Inesperadamente, nos casos em que as informações necessárias foram inseridas manualmente no contexto, a resposta do modelo apresentou uma performance arguivelmente pior do que se não estivessem presentes, alucinando a informação de que tais informações solicitadas "não existem" ou afirmando erroneamente que "não estão presentes no contexto".

Tabela 2. Exemplos de geração do modelo.

Pergunta	Resposta do Modelo
Explique o conteúdo do artigo 260 da Estatuto da Criança e do Adolescente (ECA).	O artigo 260 do Estatuto da Criança e do Adolescente (ECA) não está disponível na informação fornecida. No entanto, posso fornecer (...)
Explique o conteúdo do artigo 26 do Código Civil.	O artigo 26 do Código Civil não está presente na lista de artigos fornecida. No entanto, posso tentar ajudar com base na informação fornecida. (...)
Quais penas são esperadas pelo crime de homicídio?	A pena para o crime de homicídio no Brasil é de reclusão de 6 a 20 anos, além de multa. (...) Dica: A Lei de Drogas é uma fonte importante para entender a legislação brasileira relacionada a drogas. (...)
O que é um foguete?	Um foguete é um veículo de lançamento de satélites ou outros objetos em órbita terrestre (...) A definição não está relacionada ao Código Civil ou ao Estatuto da Criança e do Adolescente.

Fonte: Autoria própria

Por outro lado, a capacidade de dizer "não" pode ser vista como um ponto positivo para a performance do modelo, embora mesmo depois de negar, continue a gerar, eventualmente se contradizendo. Na pergunta "O que é um foguete?" da Tabela 2, é observado que o modelo manteve comportamento e informações do seu pré-treinamento, ignorando o fato de que está suposto a responder apenas com o que está presente no contexto, embora ao fim comente sobre o mesmo.

5.4. Discussão

Os experimentos revelaram desempenho insatisfatório do modelo em tarefas de pergunta–resposta jurídica baseadas em recuperação (RAG). A análise mostrou falhas recorrentes na interpretação do contexto, incapacidade de localizar informações explícitas nos trechos fornecidos e tendência a produzir respostas genéricas ou completamente desconectadas do artigo citado.

Os erros observados incluem: (i) alucinação factual, (ii) ignorar trechos relevantes, (iii) duplicação de frases, (iv) falta de baseamento no contexto e (v) respostas evasivas mesmo diante de contexto adequado. O modelo, na configuração atual, evidentemente não é adequado para a tarefa proposta, reforçando a necessidade de ajustes no *pipeline*. Os resultados, apesar de negativos, fornecem direcionamento para iterações futuras.

6. Conclusão

Os experimentos realizados evidenciam que o modelo avaliado apresenta desempenho insatisfatório no contexto de recuperação e geração de respostas baseadas em legislação brasileira. Mesmo em cenários simples, o sistema produziu respostas incorretas, alucinações factuais e inconsistências estruturais, demonstrando simultaneamente baixa capacidade de recuperar textos pertinentes e utilizar adequadamente os textos recuperados.

Esses resultados mostram que o sistema não atinge um nível de confiabilidade aceitável para aplicações jurídicas. Conclui-se, portanto, que são necessárias melhorias substanciais no processo de indexação, no mecanismo de recuperação e na calibração do modelo para que o uso desse tipo de abordagem torne-se viável.

Referências

aaaa.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Daniel Han, M. H. and team, U. (2023). Unsloth.

Douze, M., Guzhva, A., Deng, C., Johnson, J., Szilvasy, G., Mazaré, P.-E., Lomeli, M., Hosseini, L., and Jégou, H. (2025). The faiss library. *IEEE Transactions on Big Data*.

Hassani, H. and Silva, E. S. (2023). The role of chatgpt in data science: how ai-assisted conversational interfaces are revolutionizing the field. *Big data and cognitive computing*, 7(2):62.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., et al. (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Moreira, N. S., Martelli, F., MAKOWSKI, R. M., and Stumpf, A. C. (2011). Linguagem jurídica: termos técnicos e juridiquês. *Unoesc & Ciência - ACSA*, 1(2):139–146.

OpenAI (2024). Chatgpt (gpt-5.1). ⟨https://chat.openai.com⟩. Large language model used as LLM-as-a-judge.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al. (2019). Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32.

Reimers, N. and Gurevych, I. (2019a). Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084*.

Reimers, N. and Gurevych, I. (2019b). 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.

Reitz, K., Benfield, C., Cordasco, I. S., and Prewitt, N. (2025). Requests: Http for humans™. Python library, available at GitHub: https://github.com/psf/requests.

Richardson, L. (2025). Beautiful soup: Python library for pulling data out of html and xml files. Accessible via PyPI as “beautifulsoup4”.

Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M.-A., Lacroix, T., Rozière, B., Goyal, N., Hambro, E., Azhar, F., et al. (2023). Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

von Werra, L., Belkada, Y., Tunstall, L., Beeching, E., Thrush, T., Lambert, N., Huang, S., Rasul, K., and Gallouédec, Q. (2020). Trl: Transformer reinforcement learning. ⟨https://github.com/huggingface/trl⟩.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., Davison, J., Shleifer, S., von Platen, P., Ma, C., Jernite, Y., Plu, J., Xu, C., Scao, T. L., Gugger, S., Drame, M., Lhoest, Q., and Rush, A. M. (2020). Huggingface's transformers: State-of-the-art natural language processing.