UNIVERSIDADE FEDERAL DO PARANÁ

ERNANI DE SOUZA CUBAS NETO

APLICAÇÃO DE TÉCNICAS DE *Embeddings* PARA AVALIAÇÃO DE SIMILARIDADE TEXTUAL DE DEMANDAS ENCAMINHADAS AO MINISTÉRIO PÚBLICO DO ESTADO DO PARANÁ

CURITIBA, PR
2024

ERNANI DE SOUZA CUBAS NETO

APLICAÇÃO DE TÉCNICAS DE *Embeddings* PARA AVALIAÇÃO DE SIMILARIDADE TEXTUAL DE DEMANDAS ENCAMINHADAS AO MINISTÉRIO PÚBLICO DO ESTADO DO PARANÁ

Trabalho de Conclusão de Curso apresentado como requisito parcial à obtenção do título de Especialista em Inteligência Artificial Aplicada no Programa de Pós-Graduação em Inteligência Artificial Aplicada, Setor de Educação Profissional e Tecnológica, da Universidade Federal do Paraná.

Orientador: Razer Anthom Nizer Rojas Montaño.

CURITIBA, PR
2024

TERMO DE APROVAÇÃO

Os membros da Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação INTELIGÊNCIA ARTIFICIAL APLICADA da Universidade Federal do Paraná foram convocados para realizar a arguição da Monografia de Especialização de **ERNANI DE SOUZA CUBAS NETO** intitulada: **APLICAÇÃO DE TÉCNICAS DE Embeddings PARA AVALIAÇÃO DE SIMILARIDADE TEXTUAL DE DEMANDAS ENCAMINHADAS AO MINISTÉRIO PÚBLICO DO ESTADO DO PARANÁ**, que após terem inquirido o aluno e realizada a avaliação do trabalho, são de parecer pela sua aprovação no rito de defesa.

A outorga do título de especialista está sujeita à homologação pelo colegiado, ao atendimento de todas as indicações e correções solicitadas pela banca e ao pleno atendimento das demandas regimentais do Programa de Pós-Graduação.

Curitiba, 04 de Julho de 2024.

RAZER ANTHOM NIZER ROJAS MONTAÑO
Presidente da Banca Examinadora

JAIME WOJCIECHOWSKI
Avaliador Interno (UNIVERSIDADE FEDERAL DO PARANÁ)

Aplicação de Técnicas de *Embeddings* para Avaliação de Similaridade Textual de Demandas encaminhadas ao Ministério Público do Estado do Paraná

Ernani de Souza Cubas Neto
Setor de Educação Profissional e Tecnológica
Universidade Federal do Paraná
Curitiba, Brasil
ernanicubas@ufpr.br

Razer Anthom Nizer Rojas Montaño
Setor de Educação Profissional e Tecnológica
Universidade Federal do Paraná
Curitiba, Brasil
razer@ufpr.br

***Resumo*—O objetivo deste trabalho é apresentar um estudo de aplicação de técnicas de similaridade de demandas encaminhadas através de atendimento ao cidadão pelo Ministério Público do Estado do Paraná. O estudo apresenta a implantação de modelos de Processamento de Linguagem Natural, desde o pré-processamento de textos a implantação e avaliação de algoritmos de *embedding*. Foram avaliados os modelos *BERTimbau* e *Multilingual Universal Sentence Encoder*. Parte da base de dados do MPPR foi vetorizada e realizam-se testes de similaridade dos atendimentos anteriores com novas demandas encaminhadas, avaliando os resultados encontrados por estes dois modelos.**

***Palavras-chave*—Ministério Público,PLN,BERT,Codificador de Frase Universal**

***Resumo*—The objective of this work is to present a study of the application of techniques for similarity of demands sent through citizen assistance by Public Prosecutor's Office of the State of Paraná. The study presents the implementation of Natural Language Processing models, from text pre-processing to the implementation and evaluation of embedding algorithms. Models *BERTimbau* and *Multilingual Universal Sentence Encoder* were evaluated. Part of the MPPR database was vectorized and similarity tests were carried out between previous citizen assistances and new demands, evaluating the results found by these two models.**

***Index Terms*—Public Prosecutor's Office,NLP,BERT,Universal Sentence Encoder**

I. DESENVOLVIMENTO

O Ministério Publico tem como um dos seus principais objetivos a defesa dos direitos da sociedade. Para realizar esta atividade de maneira eficiente ela disponibiliza diversos meios de integração com o cidadão, através de atendimentos presenciais ou remotos. No Ministério Público do Estado do Paraná existe uma estrutura específica com esta atribuição, conduzida pela Coordenadoria da Política Estadual de Atendimento ao Público, que coordena centrais de atendimentos e promotorias espalhadas por todo o estado [1].

Independente do meio por onde a demanda da população é recebida, o cadastro realiza-se em sistema computacional único. Dessa forma, a instituição centraliza as demandas em uma mesma base de dados. Esta profusão de canais e o volume de informações cadastradas produzem alguns desafios para a instituição, como por exemplo demandas repetidas ou de mesma natureza.

Com o objetivo de facilitar a identificação de atendimentos semelhantes e diminuir o esforço ministerial na resolução dos problemas, através da análise semântica dos textos das demandas cadastradas, este trabalho apresenta a implementação de modelos NLP (*Natural Language Processing*) utilizando modelos de *embeddings*.

O Processamento de Linguagem Natural é uma área que começou a surgir na década de 1950, como a interseção entre inteligência artificial e linguística [2]. Dentre suas principais motivações destacam-se o reconhecimento pelo computador da linguagem humana e execução de tarefas que a envolvem, como analise de sentimentos, extrações de informações, classificação e similaridade de textos (sendo este o objetivo deste estudo), por exemplo [3].

Umas das técnicas utilizadas na compreensão da linguagem natural pelas máquinas é o *embedding*. *Word Embedding* refere-se a uma abordagem para representar palavras como vetores matemáticos, onde a semântica e as relações entre as palavras são capturadas pela proximidade dos vetores no espaço [4]. Existem diversos modelos de transformação de texto em *embedding*. Este documento apresenta o estudos de duas técnicas distintas: *Multilingual Universal Sentence Encoder* e *BERTimbau*.

A. Descrição dos dados

A base de dados utilizada é a existente em ambiente em produção do sistema *ePROMP*. Esta aplicação é utilizada pela área finalística do Ministério Público do Estado do Paraná para registro e acompanhamento de procedimentos extrajudiciais e atendimentos ao cidadão. Todas as informações presentes nesta base foram registradas por membros e servidores da instituição

ou encaminhadas diretamente pelo cidadão através do portal *MP Atende* [5].

A base de dados atual possui aproximadamante 350 mil registros. Optou-se, para avaliação e testes, utilizar 50 mil dados de atendimentos. Esta escolha se deve à limitação de performance da máquina utilizada nos testes (a avaliação de 50 mil vetores gera um tempo considerável, mas tolerável, de processamento).

B. Métodos

Esta seção descreve os métodos aplicados para realizar a análise de similaridade dos atendimentos. Serão ao todo duas fases: A primeira, conforme apresentado na Figura 1, apresenta a trasformação em *embedding* dos atendimentos já existentes na base de dados. Os registros são coletados, as descrições passam por um pré-processamento e posteriormente pela vetorização através de um modelo definido (*Multilingual Universal Sentence Encoder* ou *BERTimbau*) gerando na saída um tensor que é serializado e persistido em uma tabela própria.

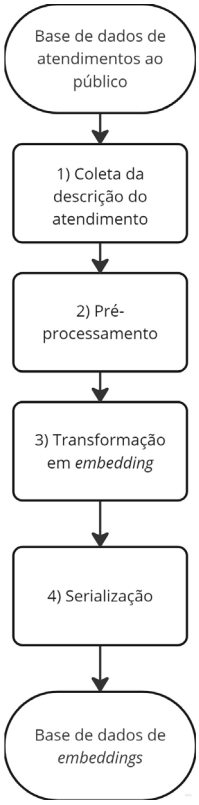

Figura 1. Etapas do método de vetorização da base de dados.
Fonte: O autor (2024)

Já a segunda fase é a comparação de um registro de entrada com a base de dados existente (Figura 2). O novo atendimento passa pelas mesmas fases apresentadas anteriormente (pré-processamento e vetorização). Para cada registro presente na base, é realizada a deserialização e aplicação de um cálculo de comparação com o *embedding* do novo registro, gerando em sua saída um *score* de similaridade.

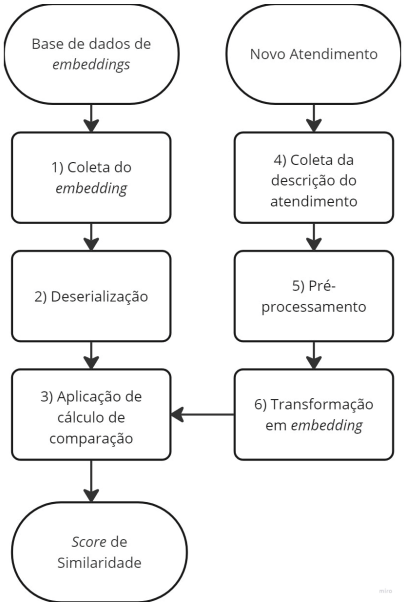

Figura 2. Etapas do método de comparação da base de dados com um novo registro.
Fonte: O autor (2024)

Para a realização dos testes, foram criados dois cenários diferentes: No primeiro, a *string* foi pré-processada, aplicou-se o modelo *BERTimbau* para a geração do *embedding* e a comparação de tensores foi realizada através da similaridade de cosseno. Para o segundo, o texto foi normalizado e aplicou-se a vetorização pelo modelo *Multilingual Universal Sentence Encoder*. Para comparação dos tensores, usou-se o cálculo do produto interno.

1) Cenário 1 - Modelo BERTimbau:

a) Coleta da descrição do atendimento e embeddings pré-gerados: Nesta etapa, os dados serão coletados diretamente da base de dados de produção. No caso, a descrição do atendimento é utilizada como *corpus*, que é a unidade de texto que corresponde à entrada do processamento de linguagem natural. A base utilizada é *Oracle*, e para que a aplicação acesse os dados dos atendimentos optou-se pelo uso da biblioteca *oracledb*.

b) Pré-processamento dos dados: Aqui, o *corpus* é preparado para sua posterior conversão em números. Dias [6] apresenta diversas técnicas de pré-processamento e extração de recursos a serem implementadas na linguagem humana, com o objetivo de facilitar a posterior conversão e interpretação pela máquina. Várias dessas técnicas são implementadas pelo pacote da linguagem *Python* NLTK (*Natural Language Tool Kit*[1]). Para os testes no primeiro cenário, foram utilizadas as seguintes técnicas:

- Segmentação: Também conhecido com *tokenização*, é o processo de quebra tanto do texto em frases, como a quebra de frases em palavras. Para este fim, usou-se a

[1]https://www.nltk.org/

classe *AutoTokenizer* da biblioteca *transformers*[2], usando um modelo pré-treinado para a língua portuguesa.

- Mudança para *lowercase*: Este processo tem o objetivo de evitar que palavras que representem o mesmo significado, mas estejam escritas de formas diferentes, tenham interpretações e criem vetores distintos.
- Exclusão de caractéres não alfanuméricos: Substitui todos os caracteres alfanúmericos, como símbolos e pontuações, em espaços em branco. Utilizou-se a biblioteca *re* nesta técnica.
- Remoção de *Stopwords*: Existem algumas palavras, como "o", "são", "é"e outras, que são consideradas irrelevantes no Processamento de Linguagem Natural. Essas palavras não acrescentam nenhum valor significativo ao sentido de uma frase e geralmente são desconsideradas em certos casos de uso específicos. No NLP, essas palavras são conhecidas como *stopwords*. A remoção dessas palavras pode ajudar a reduzir a dimensionalidade dos dados e melhorar a eficiência dos algoritmos de processamento de texto.

c) Transformação em embedding: Uma vez pré-processado o *corpus*, é necessário transformar este texto em números, ou mais especificamente em um vetor. Esta representação vetorial é denominada *embedding*. Vários são os modelos capazes de realizar esta transformação. Para o primeiro cenário utilizou-se o modelo pré-treinado para a língua portuguesa *BERTimbau*, baseado no BERT. Segue abaixo uma apresentação técnica deste modelo:

- BERT: Sigla para *Bidirectional Encoder Representations from Transformers*, este é um modelo de representação de linguagem baseado na técnica denominada *Transformers*, considerado o estado da arte em Processamento de Linguagem Natural [7]. O *Transformer* é uma arquitetura inovadora no Processamento de Linguagem Natural que aborda desafios de dependências de longo alcance em tarefas sequência-a-sequência. Em vez de depender de Redes Neurais Recorrentes ou convolução, o *Transformer* se baseia exclusivamente na autoatenção (*Self-Attention*) para calcular as representações de entrada e saída. Esse modelo, introduzido em 2017, utiliza o mecanismo de atenção para ponderar a importância de diferentes partes dos dados de entrada, sendo amplamente aplicado em tarefas de PLN e também explorado em outros domínios, como compreensão de vídeo [8]. BERT foi projetado para pré-treinar representações profundas bidirecionais a partir de texto não rotulado, condicionando simultaneamente tanto o contexto à esquerda quanto à direita em todas as camadas. Como resultado, o modelo pré-treinado BERT pode ser ajustado com apenas uma camada adicional de saída para criar modelos de última geração para uma ampla variedade de tarefas, como resposta à perguntas e inferência de linguagem, sem modificações substanciais na arquitetura específica da tarefa. BERT é conceitualmente simples e empiricamente poderoso [9].
- *BERTimbau*: O *BERTimbau* é um modelo BERT pré-treinado para o português do Brasil, desenvolvido com o objetivo de melhorar o desempenho em tarefas de processamento de linguagem natural. O modelo é treinado em dados não rotulados, utilizando artigos da Wikipedia e milhões de páginas da web do *corpus* brWaC. Após o pré-treinamento, o *BERTimbau* é ajustado finamente em três tarefas de PLN: similaridade textual de sentenças, reconhecimento textual de implicação e reconhecimento de entidades nomeadas. Os resultados mostram que o *BERTimbau* supera modelos multilíngues e abordagens anteriores em português, alcançando desempenho de ponta em todas as tarefas. Além disso, o tamanho do modelo BERT (Base vs. Large) impacta no desempenho, com o modelo Large alcançando melhores resultados. O *BERTimbau* também é avaliado em uma abordagem de *embeddings* contextuais, mostrando-se eficaz em cenários com recursos limitados. Os autores disponibilizam o *BERTimbau* para a comunidade, visando fornecer bases sólidas para pesquisas futuras em PLN [10].

d) Serialização e Deserialização: Para guardar na base de dados um tensor gerado anteriormente, é necessário transformá-lo em texto. No processo de serialização, o *embedding* é primeiro convertido em um *array numpy*, e depois em uma *string* JSON. Já a deserialização faz o oposto, convertendo o JSON em tensor novamente, para o processo de comparação da base com o novo atendimento. No projeto usou-se a biblioteca *json* para realizar estas conversões.

e) Métricas de avaliação: Para encontrar os escores de similaridade, existem várias métricas que medem as distâncias entre vetores. Aqui, usou-se a similaridade cosseno, que mede a similaridade entre dois vetores de um espaço de produto interno de maneira geral. Ela mede o cosseno do ângulo entre dois *embeddings* e determina se eles estão apontando aproximadamente na mesma direção ou não. Quando os *embeddings* estão apontando na mesma direção, o ângulo entre eles é zero, então sua similaridade cosseno é 1. Quando os *embeddings* são perpendiculares entre si, o ângulo entre eles é de 90 graus e a similaridade cosseno é 0. Por fim, quando o ângulo entre eles é de 180 graus, a similaridade cosseno é -1. Desta forma, o valor encontrado encontra-se entre -1 e 1, onde 1 representa maior similaridade e -1 menor similaridade. A equação é representada em (1).

$$s(A,B) = \frac{A * B}{\|A\| * \|B\|} = \frac{\sum_{i=1}^{n} A_i * B_i}{\sqrt{\sum_{i=1}^{n} A_i^2} * \sqrt{\sum_{i=1}^{n} B_i^2}} \quad (1)$$

2) Cenário 2 - Modelo Multilingual Universal Sentence Encoder: Serão apresentadas abaixo as etapas que se distinguem do primeiro cenário.

a) Pré-processamento dos dados: Conforme citado em [11] sobre o modelo *Universal Sentence Encoder*, usado neste cenário, o pré-processamento já é realizado na entrada de texto, não sendo necessário pré-processar os dados antes de aplicar o módulo. Porém, antes de gerar o *embedding*, o texto de entrada

[2]https://pypi.org/project/transformers/

passou pelo processo de normalização (letras em minúsculo e retirada de caracteres não alfa-numéricos).

b) Transformação em embedding: Aqui, optou-se pelo *Multilingual Universal Sentence Encoder*. Este modelo é baseado nas arquiteturas *Transformer* e CNN, e oferece desempenho competitivo com o estado da arte em: recuperação semântica (SR), recuperação de bitexto de pares de tradução (BR) e resposta a perguntas de recuperação (ReQA). Esses modelos incorporam texto de 16 idiomas em um único espaço semântico usando um codificador duplo treinado em várias tarefas que aprende representações vinculadas usando tarefas de ponte baseadas em tradução. O *Transformer* (também presente no modelo BERT) usa atenção bidirecional para calcular representações conscientes do contexto de *tokens* em uma frase, levando em consideração tanto a ordem quanto a identidade dos *tokens*. Já o modelo de codificação de sentenças CNN alimenta as incorporações de sequência de *tokens* de entrada em uma rede neural convolucional [12].

c) Métricas de avaliação: para contabilizar o *score* na comparação entre os tensores, usou-se o produto interno de vetores. Esta é uma operação que combina os elementos correspondentes dos vetores e soma os resultados. A equação (2) representa este cálculo.

$$p(A, B) = \sum_{i=1}^{n} A_i * B_i \tag{2}$$

C. Tecnologias

O *script* foi executado em um computador DELL com as seguintes especificações:

- Processador Intel(R) Xeon(R) W-1270 CPU @ 3.40GHz
- RAM instalada 16,0 GB
- Sistema Operacional Windows 10 Pro for Workstations

A base de dados usada é Oracle versão 12c Enterprise Edition

Para o desenvolvimento do projeto, foram utilizadas as seguintes aplicações e bibliotecas:

- Linguagem Python, versão 3.11.2;
- Visual Studio Code, versão 1.77.3;
- Pacote *nltk*, versão 3.8.1;
- Pacote *TensorFlow*, versão r2.11;
- Pacote *transformers*, versão 4.41.2;
- Pacote *torch*, versão 2.3.0;
- Pacote *scipy*, versão 1.10.1;
- Pacote *numpy*, versão 1.24.1;
- Pacote *oracledb*, versão 2.2.1;
- Pacote *fastapi*, versão 0.103.2.

II. RESULTADOS E DISCUSSÕES

Devido a natureza do estudo, a avaliação dos resultados obtidos em ambos os cenários foi realizada de maneira empírica, observando, para uma mesma frase de entrada, os atendimentos da base de dados considerados similares. Os testes foram realizados utilizando o pacote *fastapi*[3]. O serviço recebeu como entrada uma *string*, e retornou um *json* contendo os 5 atendimentos com melhor *score* de similaridade, contendo o *id* do registro de atendimento, o valor do *score* e a *string* com a descrição do atendimento. Foram criados dois serviços em separado, para cada cenário. No primeiro teste usou-se a seguinte descrição: "Solicitante pede informação sobre aquisição de remédio gratuito". Anteriormente, ao avaliar a base de dados, verificou-se que, em muitas das demandas desta natureza, os usuários optam por usar a palavra "medicamentos". A ideia então era realizar uma avaliação semântica dos modelos. Em ambas as situações, tanto no *Universal Sentence Encoder Multilingual* como no *BERTimbau* os resultados foram satisfatórios. Eles podem ser visualizados na Tabela I.

O segundo teste teve o mesmo tipo de avaliação. A descrição de entrada era: "o atendido busca contribuição alimentar para seu neto". O objetivo era que ambos os modelos entregassem como resultado atendimentos relativos à pensão alimentícia. Aqui, os resultados também foram os esperados.Porém, observou-se que o atendimento de maior *score* entregue pelo modelo *BERTimbau* nada tinha que ver com o assunto de entrada, onde a descrição do fato estava relacionada com uma solicitação de orientação para obter auxílio emergencial. Os resultados estão na Tabela II.

Já para o terceiro teste, o objetivo foi diferente. A ideia nesta situação era avaliar o comportamento dos modelos em frases de entrada com erros gramaticais. Para isso, propositadamente, foi enviado aos dois modelos a frase "averiguassão de patenidade". Percebe-se que a primeira palavra possui erro na escrita, e a segunda uma omissão de letra. Mesmo assim, o modelo *Universal Sentence Encoder Multilingual* conseguiu entregar nos seus primeiros resultados atendimentos com o mesmo contexto (averiguação de paternidade). Já o modelo *BERTimbau* não conseguiu tratar os erros da *string* de entrada e apresentou resultados distorcidos. Ambas as saídas dos modelos podem ser verificadas na Tabela III (Por questão de privacidade, nomes presentes nos resultados foram omitidos).

Conclui-se neste estudo que o uso destes modelos, destacando-se o *Universal Sentence Encoder Multilingual*, pode gerar resultados satisfatórios e de acordo com o objetivo deste trabalho. Futuramente, a expectativa é que este projeto torne-se um módulo do sistema finalístico da instituição, e entregue aos seus usuários, assim que uma nova demanda seja cadastrada, uma lista de registros similares, e assim o guie na conclusão do atendimento.

Como trabalho futuro, pretende-se analisar e avaliar o resultado obtido em ambos os modelos por meio de métricas estatísticas robustas, com a finalidade de comprovar a eficácia deste estudo. Para isso, serão estudadas maneiras de avaliação específicas para PLN que determinem com precisão a qualidade dos modelos, para posterior implantação deste projeto no Ministério Público do Paraná.

Além de avaliar novas maneiras de metrificar resultados, é importante também entender o porquê de atendimentos com natureza diferente das demandas de entrada receberam *scores* altos, conforme apresentado nos testes 2 e 3 para o modelo *BERTimbau*.

[3]https://fastapi.tiangolo.com/

Outra atividade a ser desenvolvida antes da implantação deste projeto é a realização de testes internos com avaliação humana, onde profissionais que atuam no atendimento ao público possam analisar os resultados apresentados e assim mensurar a qualidade do modelo.

REFERÊNCIAS

[1] MPPR, "Atendimento - ministério público do paraná." Disponível em: <https://site.mppr.mp.br/atendimento/>. Acesso em: 10 de jun. de 2024.

[2] P. M. Nadkarni, L. Ohno-Machado, and W. W. Chapman, "Natural language processing: an introduction," *Journal of the American Medical Informatics Association*, vol. 18, p. 544–551, 2011.

[3] J. S. Martins *et al.*, *Processamentos de linguagem natural*. Porto Alegre: SAGAH, 1 ed., 2020.

[4] D. Karani, "Introduction to word embedding and word2vec." Disponível em: <https://towardsdatascience.com/introduction-to-word-embedding-and-word2vec-652d0c2060fa/>. Acesso em: 03 de jun. de 2023.

[5] MPPR, "Mp atende - formulário." Disponível em: <https://site.mppr.mp.br/atendimento/Pagina/MP-Atende-Formulario/>. Acesso em: 03 de jun. de 2023.

[6] A. da S. Dias, *Processamentos de linguagem natural*. São Paulo: Platos Soluções Educacionais S.A, 1 ed., 2021.

[7] D. S. Academy, "Transformadores – o estado da arte em processamento de linguagem natural." Disponível em: <https://www.deeplearningbook.com.br/transformadores-o-estado-da-arte-em-processamento-de-linguagem-natural/>. Acesso em: 13 de abr. de 2024.

[8] D. S. Academy, "Deep learning book." Disponível em: <https://www.deeplearningbook.com.br/>. Acesso em: 06 de jun. de 2023.

[9] D. Jacob, C. Ming-Wei, L. Kenton, and T. Kristina, "Bert: Pre-training of deep bidirectional transformers for language understanding," *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, vol. 1, p. 4171–4186, 2019.

[10] F. Souza, R. Nogueira, and R. de Alencar Lotufo, "Bert models for brazilian portuguese: pretraining, evaluation and tokenization analysis," *Intelligent Systems, 9th Brazilian Conference*, vol. 1, 2020.

[11] Google, "Universal sentence encoder - encoder of greater-than-word length text trained on a variety of data." Disponível em: <https://www.kaggle.com/models/google/universal-sentence-encoder>. Acesso em: 25 de nov. de 2023.

[12] Y. Yang, D. Cer, A. Ahmad, M. Guo, J. Law, N. Constant, G. H. Abrego, S. Yuan, C. Tar, Y. hsuan Sung, B. Strope, and R. Kurzweil, "Multilingual universal sentence encoder for semantic retrieval," *Association for Computational Linguistics*, vol. 1, p. 87–94, 2020.

Tabela I
PRIMEIRO RESULTADO DE TESTES DOS MODELOS

Texto de entrada: Solicitante pede informação sobre aquisição de remédio gratuito		
Modelo: "Universal Sentence Encoder Multilingual"		
id	**score**	**Descrição**
78262	0.72	"solicita informações sobre a disponibilização de medicamentos em ação judicial."
85538	0.71	"a atendida solicitou informações sobre seu pedido de medicamentos"
95540	0.71	"A atendida solicitou informações quanto ao fornecimento do medicamento - NIVOLUMABE"
104325	0.7	"A atendida solicitou informações quanto ao fornecimento do medicamento Venvanse."
58174	0.7	"solicita informações quanto ação para liberação de vaga"
Modelo: "BERTimbau"		
id	**score**	**Descrição**
20806	0.92	"solicita informações sobre pedido do medicamento spiriva."
116365	0.92	"Solicita informação referente a liberação de medicamento canabidiol, porém, não foi prescrito por médico."
8290	0.91	"Solicita informações com o intuito de obter medicamento não formicido pelo SUS."
52485	0.91	"Solicita o fornecimento do medicamento PROLIA"
80808	0.91	"A representante solicita informações para ingressar com pedido judicial do medicamento Forxiga."

Fonte: O autor (2024)

Tabela II
SEGUNDO RESULTADO DE TESTES DOS MODELOS

Texto de entrada: o atendido busca contribuição alimentar para seu neto		
Modelo: "Universal Sentence Encoder Multilingual"		
id	**score**	**Descrição**
105950	0.65	"O atendido entrou em contato para buscar orientações sobre pensão alimentícia."
57545	0.62	"Informou que tinha interesse em interpor ação visando pensão alimentícia."
66673	0.61	"A atendida pediu orientações acerca de pensão alimentícia."
89687	0.61	"A atendida solicitou informações sobre pensão alimentícia."
119059	0.6	"busca informações sobre o pensão alimentícia"
Modelo: "BERTimbau"		
id	**score**	**Descrição**
13153	0.87	"Atendida relata que seu filho de 18 anos, completados em abril de 2020, não conseguiu auxílio emergencial. Que já foi na Assistência Social mas que não obteve solução. Solicita orientação. "
61322	0.86	"a atendida informou que o genitor não paga alimentos regularmente aos filhos"
63028	0.86	"a atendida informou que o genitor não paga alimentos regularmente aos filhos"
76	0.86	"A atendida relata que é avó de menor de idade e busca pensão alimentícia para seu neto."
58288	0.86	"a atendida informou que o genitor não paga alimentos regularmente aos filhos"

Fonte: O autor (2024)

Tabela III
TERCEIRO RESULTADO DE TESTES DOS MODELOS

Texto de entrada: averiguassão de patenidade		
Modelo: "Universal Sentence Encoder Multilingual"		
id	**score**	**Descrição**
94886	0.49	"Averiguação Oficiosa de Paternidade"
98127	0.45	"Termo de Declaração - Averiguação de Paternidade"
72862	0.44	"Orientação sobre reconhecimento de paternidade."
103230	0.43	"Solicitação de certidões."
103216	0.43	"Indicação de suposto genitor"
Modelo: "BERTimbau"		
id	**score**	**Descrição**
27834	0.84	"********. Pessoa com transtorno mental. Assuntos desconexos."
51267	0.83	"Sofrendo com ameça"
12171	0.82	"********. Reclamar do Air Bnb na pandemia."
84434	0.81	"********. Pessoa com transtorno mental. Assuntos desconexos."
40392	0.81	"********. Pessoa com transtorno mental. Assuntos desconexos."

Fonte: O autor (2024)