UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Uso de Aprendizado de Máquina para Determinar as Melhores Práticas de Implementação de *Chatbots*

Danielle Christina Fernandes Inhesta

Trabalho de Conclusão de Curso
MBA em Inteligência Artificial e Big Data

ICMC USP SÃO CARLOS

UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Uso de Aprendizado de Máquina para Determinar as Melhores Práticas de Implementação de *Chatbots*

Danielle Christina Fernandes Inhesta

USP - São Carlos
2022

Danielle Christina Fernandes Inhesta

Uso de Aprendizado de Máquina para Determinar as Melhores Práticas de Implementação de *Chatbots*

Trabalho de conclusão de curso apresentado ao Departamento de Ciências de Computação do Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo - ICMC/USP, como parte dos requisitos para obtenção do título de Especialista em Inteligência Artificial e Big Data.

Área de concentração: Inteligência Artificial

Orientador: Prof. Dr. Ricardo Rodrigues Ciferri

USP - São Carlos

2022

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

F35u

Fernandes Inhesta, Danielle Christina
Uso de Aprendizado de Máquina para Determinar as Melhores Práticas de Implementação de Chatbots / Danielle Christina Fernandes Inhesta; orientadora Ricardo Ciferri. -- São Carlos, 2022.
35 p.

Trabalho de conclusão de curso (MBA em Inteligência Artificial e Big Data) -- Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2022.

1. . I. Ciferri, Ricardo, orient. II. Título.

Bibliotecários responsáveis pela estrutura de catalogação da publicação de acordo com a AACR2:
Gláucia Maria Saia Cristianini - CRB - 8/4938
Juliana de Souza Moraes - CRB - 8/6176

DEDICATÓRIA

Ao meu esposo pela compreensão e carinho.

Aos meus pais (in memoriam) que sempre me ensinaram o valor dos estudos

AGRADECIMENTOS

Agradeço ao meu orientador pela paciência e por me motivar a buscar coisas novas que fizeram a diferença nesse trabalho.

Agradeço também a coordenação e aos professores deste curso que sempre me apoiaram quando necessário e levarei o conhecimento para a minha vida profissional.

Mais uma vez, obrigada!

RESUMO

Fernandes Inhesta, Danielle Fernandes Título: Uso de Aprendizado de Máquina para Determinar as Melhores Práticas de Implementação de *Chatbots*. 2022. XX f. Trabalho de conclusão de curso (MBA em Inteligência Artificial e Big Data) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2020.

O foco principal deste trabalho de conclusão de curso foi a ampliação do conhecimento e desenvolvimento em *chatbots* que utilizam Inteligência Artificial e Aprendizado de Máquina. A ideia principal é a criação de um *chatbot* de recuperação (*Self-Retrieval*) utilizando algoritmos de normalização de dados e análise de estruturas de Processamento de Linguagem Natural em Python como: Scikit-learn, NLTK e Numpy. Como resultado foi criado o *chatbot* de nome *Pangea* que utiliza os algoritmos mencionados, utilizando 2 redes neurais: uma para classificação dos dados de mineração (similaridade) e outra para separação de palavras no *Bag of Words.* Após treinamento dos dados o chatbot é capaz de responder perguntas relacionadas ao conflito na Ucrânia sobre os temas como: óleo, gás, impacto de preços do trigo, crise nuclear, inflação etc. de forma estática.

Palavras-chave: *chatbot; natural language processing; artificial intelligence; algorithms; Self-Retrieval chatbot, machine learning, project management;*

ABSTRACT

Fernandes Inhesta, Danielle Fernandes Título: Use of Machine Learning to Determine Best Pratices to Implement Chatbots. 2022. XX f. Trabalho de conclusão de curso (MBA em Inteligência Artificial e Big Data) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2020.

The focus of this work will be the expansion of knowledge and development in chatbots that use Artificial Intelligence and Machine Learning and their history, since the 1050s when the first studies related to NLP, Turing test, etc. The main idea is to create a self-retrieval *chatbot* using data normalization algorithms and analysis of Natural Language Processing structures in Python such as: Scikit-learn, NLTK e Numpy. As a result, the chatbot named *Pangea* was created, which uses the algorithms mentioned above, using 2 neural networks: one for classification of data minning (look for similarity) and another for word separation used by Bag of Words. After training the data, the chatbot can answer questions related to the conflict in Ukraine on topics such as: oil, gas, wheat price impact, nuclear crisis, inflation, etc. in a static way.

Keywords: *chatbot; natural language processing; artificial intelligence; algorithms; Self-Retrieval chatbot, machine learning, project management;*

LISTA DE ABREVIATURAS E SIGLAS

NLP - *Natural Language Processing*

NLTK - *Natural Language Toolkit*

AI - *Artificial Intelligence*

IA - Inteligência Artificial

AM - Machine Learning

Relu - *Rectified Linear Unit*

SGD - *Stochastic Gradient Descent*

MLP - *Multi-Layer Perceptron*

SUMÁRIO

1 INTRODUÇÃO

O uso de Inteligência Artificial (IA) e Aprendizado de Máquina (AM) tem crescido rapidamente nos últimos anos e as empresas de serviços vem se adaptando de forma gradual. No entanto, as empresas ainda não estão conseguindo se adaptar na mesma velocidade de seu crescimento. IA e AM são assuntos em destaque na academia e nas empresas por uma razão bem simples: esses assuntos estão transformando radicalmente o mundo. Devido a quão promissoras são as técnicas e tecnologias decorrentes de IA e AM e da quantidade de benefícios que elas já oferecem, muitas empresas estão dispostas a usá-las para a transformação de seus negócios [10].

O *chatbot* é conhecido por muitos nomes no mundo atual. Ele pode ser chamado de *smartbot*, *chatterbot*, *talkbot*, agente interativo, agente de conversação, entidade conversacional artificial ou simplesmente *bot*. Um *chatbot* pode ser descrito como um programa desenvolvido ou uma criação humana de IA que usa várias tecnologias para imitar uma conversa que um humano teria com outro humano. A comunicação pode ocorrer via conversas de áudio ou por texto [17].

Os *chatbots* são concebidos e desenvolvidos de forma a simular a forma como um ser humano se comporta em uma conversa. Os *chatbots* são comumente usados em sistemas de diálogo, como exemplo na conversa de um cliente com um atendente de uma empresa na aquisição de serviços ou informações, que são duas das aplicações mais práticas no mundo de hoje [17].

Recentemente, *chatbots* estão sendo desenvolvidos com base em sistemas complexos que usam Processamento de Linguagem Natural (PLN) para seu processamento, em comparação com os sistemas tradicionais de *chatbot*, que procuram palavras-chave quando a entrada é fornecida e verificam a resposta que contém as palavras-chave mais correspondentes, ou um padrão de palavras, em um banco de dados [17].

1.1 Motivação

A motivação para o desenvolvimento desta pesquisa de TCC decorre das fortes demandas de mercado na solicitação de profissionais como cientistas de dados e profissionais com conhecimentos em inteligência artificial, onde projetos de implementação terão de ser executados de forma rápida e objetiva com metodologias adequadas.

Alguns setores que apresentam grande demanda atualmente por projetos de inteligência artificial são: setor médico, financeiro, automotivo, transporte público, tecnologia, mídia e comunicações [6].

A motivação para o desenvolvimento desta pesquisa é no sentido de aprofundar os conhecimentos em *chatbots* criando um *chatbot* baseado em recuperação (*Retrieval-based)* utilizando um *corpus* do *The Guardian* em inglês, retornando resultados relacionados a guerra na Ucrânia do último mês.

Os resultados serão exibidos na ferramenta *Orange 3*, importando o código para o módulo *Python Script* para melhorar a representação gráfica das perguntas e respostas.

1.2 Objetivos

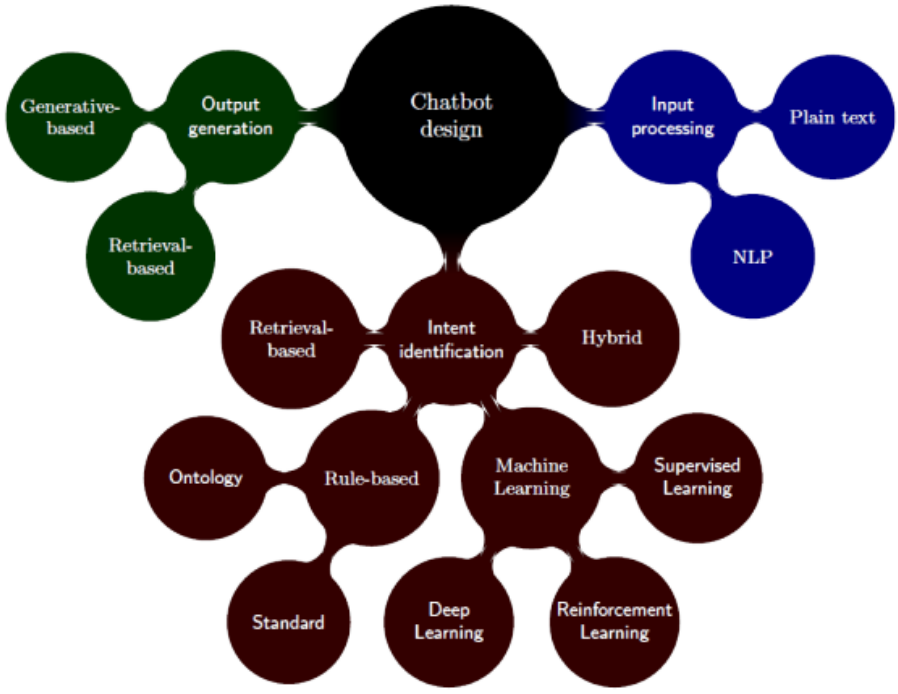
A proposta deste trabalho será a investigação mais aprofundada em implementações de *chatbots* utilizando IA e AM e como resultado a criação de um *chatbot* de recuperação (*Retrieval-based)* utilizando um *corpus* do *The Guardian*, seguindo os processos pesquisados até o momento e aprendendo de forma construtiva como um cientista de dados e gerente de projetos.

2. TRABALHOS RELACIONADOS

Segundo D´Avila [18] as abordagens de *chatbot* consistem em uma combinação de técnicas de projeto categorizadas de acordo de como sua entrada é processada; a intenção da entrada é identificada e a saída é gerada.

O Processamento de Linguagem Natural oferece algumas opções de processamento possíveis, como letras minúsculas, pontuação descartável, acentos e / ou caracteres especiais. Listas de substituições também são utilizadas como forma de normalização com uma tabela de banco de dados ou um arquivo de abreviações e coloquialismos. Métodos mais complexos de PNL para normalização, *stemming*, marcação de POS, analisadores de palavras / frases também podem ser usados [18].

As abordagens híbridas propõem um método que combina *bots* baseados em recuperação (Retrieval-based) com máquina recuperando respostas de representações vetoriais distribuídas das entradas (*embeddings*). [18]



Estrutura de um Chatbot (2017)

Abordagens baseadas em um *chatbot* (Retrieval-based) contam com algoritmos de busca para recuperar a resposta do *bot*. Isso permite o uso de dados estruturados e / ou não

estruturados como fontes para uma resposta sem a necessidade de combinar a intenção com uma mensagem ou padrão conhecido anteriormente. [18] Support Vector Machine (SVM) e classificadores de Regressão Logística (algoritmos de Aprendizagem Supervisionada) identificam a intenção em mensagens do usuário.

O Aprendizado por Reforço simulando diálogos entre dois agentes com um modelo de sequência para sequência (seq2seq). Artificial Redes Neurais (ANN) também podem ser usadas para calcular o nível de confiança de uma intenção ou uma resposta em um conjunto de dados dada a entrada. Neste trabalho o autor explica quais foram os classificadores, algoritmos e o *design* aplicado para implementação de um *chatbot Self-Retrieval* chamado KINO que responde perguntas sobre filmes utilizando uma base de dados local.

Segundo Oberdan Almeida Junior [19], Processamento de Linguagem Natural (PLN) é uma subárea de pesquisa da Inteligência Artificial (IA) que estuda a comunicação, por meio da linguagem natural, entre o Homem e o Computador. De um modo mais formal, PLN pode ser definida como um ramo da IA que tem por objetivo analisar, interpretar ou produzir texto em uma língua natural.

PLN possibilita que os seres humanos se comuniquem com os computadores de uma forma mais natural, utilizando sua língua nativa. O seu uso permite que os usuários não precisem aprender uma linguagem artificial para realizar a interação com o computador, a qual a sintaxe costuma ser um pouco mais difícil, como as linguagens de programação e de consulta de banco de dados. Para ser considerado um sistema baseado em PLN, ele deve atender a duas condições: [19]

• Uma parte do sistema deve ser codificado em linguagem natural;

• O processamento de entrada e/ou saída é baseado em aspectos sintáticos, semânticos e/ou pragmáticos de uma língua natural.

Alan Turing e a relação com os *chatbots*:

Sobre a história da linguagem natural, o autor menciona Alan Turing que propõe considerar a seguinte questão: “As máquinas podem pensar?”. Para melhor refletir sobre esse questionamento, Turing sugere um jogo chamado *The Imitation Game* (Jogo da Imitação). Posteriormente, esse jogo ficou conhecido como o Teste de Turing. O trabalho de Turing possibilitou o surgimento de máquinas capazes de dialogar com o ser humano usando linguagem natural. Os primeiros sistemas que interagiam com o usuário em linguagem natural datam do início da década de 1960. Esses sistemas são intitulados de Sistemas de Pergunta-

Respostas. Eles recebem uma pergunta em linguagem natural e, por meio de pesquisa em uma base de dados que contém pares Perguntas / Respostas, retornam uma resposta. [19]

Outro tipo de sistema que também surgiu a partir do trabalho de Turing foi o *chatterbot*. ELIZA, criada por Weizenbaum em 1966, foi o primeiro sistema desse tipo registrado na literatura. O termo *chatterbot* foi introduzido apenas em 1994 por Michael L. Mauldin. Em seu trabalho, Mauldin descreve a criação e atuação dos *chatterbots* no jogo TINYMUD, no qual eles se faziam passar por jogadores reais. Em novembro de 1991, o primeiro Teste de Turing foi realizado em grande escala no *Boston Computer Museum*, nos Estados Unidos, esse evento é conhecido como *The Loebner Prize14*. O evento, que recebe o nome de um dos seus fundadores, Dr. Hugh Loebner, e acontece anualmente desde o seu início. Ele tem o objetivo de reconhecer e premiar o *chatbot* considerado mais "humano" pelos juízes. Para vencer este concurso, o *chatbot* deve conseguir enganar metade dos juízes (o Teste de Turing original fala somente em enganar mais de 30% dos juízes). Desde a instituição do concurso, nenhum *chatbot* conseguiu esta proeza de enganar a metade dos juízes, restando apenas premiar o *chatbot* que apresentou o melhor desempenho. *Chatbots* como Agentes Inteligentes Conversacionais é uma entidade que pode perceber o ambiente por meio de sensores e agir sobre ele por intermédio de atuadores. Definido dessa forma, um agente pode ser uma entidade humana ou artificial (e.g., robôs, softwares). Um agente humano percebe o ambiente por meio dos seus órgãos (e.g., olhos, ouvidos) e age sobre ele usando seus atuadores (e.g., mãos, pernas). Um agente robô pode ter câmeras e/ou infravermelho como sensores, e motores como atuadores. Já os agentes de *software*, também conhecidos como agentes virtuais, podem perceber o seu ambiente por meio de uma sequência de teclas para digitação, por meio de arquivos e pacotes de redes, ou mesmo por voz (reconhecimento de voz), e pode atuar no ambiente por intermédio de um texto exibido na tela, escrevendo algo em um arquivo ou mesmo falando (síntese de voz). Para um agente de *software* ser considerado inteligente, ele deve conseguir agir de forma autônoma em um ambiente, a fim de cumprir seus objetivos de forma satisfatória. Os *chatbots* são projetados para perceber a entrada do usuário (percepção) por meio de uma interface (ambiente) e oferecer uma resposta adequada (ação), buscando manter um diálogo coerente com o usuário. Devido a essas características, os *chatbots* podem ser considerados como Agentes Inteligentes (AI). Esses agentes têm como característica principal a capacidade de conversar com os usuários, sendo então chamados de Agentes Inteligentes Conversacionais (ou simplesmente Agentes Conversacionais (ACs). O diálogo foi uma característica proposta como um meio de testar os primeiros conceitos de IA, que surgiu a partir do trabalho de Turing.

Neste trabalho o autor explica a história e o desenvolvimento de um *chatbot* chamado “Beck” utilizando a ferramenta *ChatScript* que é capaz de interagir com adolescentes com depressão, explicando os conceitos de PLN e como foram adicionados “sentimentos” nas conversas, percentual de aceitação e testes.

A tarefa de Compreensão da Língua Natural (NLU) pode ser considerada como o processo de tradução da linguagem natural para uma linguagem interpretável por um computador. Esta seção explica o estado da arte em sistemas NLU que realizam esta tradução (incluindo a representação da lógica predicada) [20].

O Entendimento da Linguagem Natural (NLU) por computadores começou em 1950 como uma disciplina relacionada à linguística. Isso evoluiu para incorporar aspectos de muitas outras disciplinas (como inteligência artificial e lexicografia). Uma boa maneira de definir a NLU é considerar diferentes aplicações que as pesquisas abordam. Essas aplicações podem ser divididas em duas classes amplas: (1) aplicativos baseados em texto e (2) aplicativos baseados em diálogo. [20]

1. Os aplicativos baseados em texto envolvem o processamento de texto escrito, como livros, diários, relatórios, manuais e muito mais. Esta classe de aplicações são sistemas focados em encontrar informações apropriadas, extração de informações, tradução automática e soma automática.

2. As aplicações baseadas no diálogo envolvem a comunicação homem-máquina. Normalmente, esses aplicativos incluem sistemas como perguntas e respostas, atendimento pessoal por telefone e tutoria automatizada

Neste trabalho o autor explica o conceito da utilização de *PLN* e explica como os algoritmos entendem o processo de linguística: semântica, sintaxe, morfologia de palavras, análise de sentimentos e discurso utilizados no desenvolvimento de *chatbots.*

3 FUNDAMENTAÇÃO TEÓRICA

Os *chatbots* estão entre as aplicações corporativas mais tangíveis da Inteligência Artificial. De assistentes *on-line*, como o Cortana, da Microsoft, a "*bots* auxiliares" em aplicativos de mensagens, como o *Whatsapp,* passando por aplicativos domésticos, como o Alexa da Amazon e Siri.

Chatbots podem ser utilizados para diversas funcionalidades como controlar aparelhos, encontrar restaurantes, fazer chamadas, responder perguntas tanto utilizando canais de voz ou via chat. A infraestrutura tecnológica dos *chatbots* evolui continuamente, assim como as possibilidades de integração com as demais plataformas de comunicação e sistemas legados.

Os *chatbots* em sua essência são baseados em Processamento de Linguagem Natural onde imitam uma conversa humano-humano. Mas para que isso ocorra se faz necessário o tratamento da base de dados utilizada (*corpus*) para que ele consiga aprender de forma adequada. Neste processo é essencial considerar o tratamento dos dados (normalização) antes de treinar os modelos e efetuar vários testes e ajustes para aumentar a sua acurácia.

Os modelos de aprendizado em PLN que serão utilizados nesta pesquisa serão as bibliotecas do Python NLTK, Scikit-learn e Numpy.

Há dois tipos principais de *chatbots* que serão investigados na pesquisa:

- *Chatbot Gerativo (Generative chatbot):* Dependendo do modelo Gerativo, este tipo de chatbot não usa nenhum repositório de conversação. É uma forma avançada de chatbot empregando aprendizado de máquina para responder consultas de usuários. A maioria dos chatbots modernos funcionam de acordo com o modelo gerativo. Portanto, eles podem responder a quase todos os tipos de perguntas. Além disso, eles cobrem o quociente humano em suas conversas. Portanto, um chatbot com aprendizagem profunda é mais adaptável às consultas de seus clientes, mas não deve ser confundido em imitar o pensamento humano em seus padrões de conversação [9]. O Modelo gerativo, aprendido nas aulas também tem como propósito em adicionar ruído nos dados para treinar o identificador para separação de dados verdadeiros e falsos.

- *Chatbot* baseado em recuperação (*Retrieval-based chatbot*): Este tipo de *chatbot* usa um repositório pré-definido para responder consultas. Estes só podem responder a um limitado conjunto de perguntas pré-definidas e podem fornecer a mesma resposta para duas perguntas diferentes. No entanto, uma limitação deste tipo de *chatbot* é a necessidade de se selecionar o

> sistema de resposta para o *chatbot* responder. Um *chatbot* baseado em modelo de recuperação comete menos erros, uma vez que consulta um banco de dados para perguntas do usuário e fornece respostas de acordo. Porém, não pode responder a determinadas perguntas que não constam no repositório e pode não parecer humano, sendo mais facilmente perceptível como robótico. A maioria dos sites disfarça um *chatbot* de recuperação como um *livechat*, já que eles são simples de codificar e criar. Embora tradicional, tal *chatbot* acompanha as mensagens anteriores de um usuário, mas só pode responder perguntas que são simples e não responde a consultas complexas [9].

A NLTK é um conjunto de bibliotecas de construção de programas em Python para trabalhar com dados de processamento de linguagem humana ou linguística computacional. Ele oferece suporte as funcionalidades de classificação, linguística empírica, ciências cognitivas, inteligência artificial, recuperação de informações e aprendizado de máquina que são utilizados pelas bibliotecas NLP, e utiliza bibliotecas em várias línguas inclusive português. Ele foi desenvolvido por Steven Bird e Edward Loper no Departamento de Ciência da Computação e da Informação da Universidade da Pensilvânia.

A biblioteca sklearn Scikit-learn (anteriormente scikits.learn e também conhecida como sklearn) é uma biblioteca de aprendizado de máquina de *software* livre para a linguagem de programação Python. Esta biblioteca apresenta vários algoritmos de classificação, regressão e agrupamento, incluindo vetores, random forest, aumento de gradiente, k-means e DBSCAN, sendo projetada para interoperar com as bibliotecas numéricas e científicas do Python NumPy e SciPy.

4 METODOLOGIA

As metodologias de pesquisa serão feitas por meio de procura de artigos relacionados ao assunto investigado nessa pesquisa de TCC nas seguintes bibliotecas digitais: Google Scholar, DBLP, Scopus, Web of Science, Springer Link, ACM Digital Library, IEEE Xplore, SciELO, Portal de Periódicos da CAPES, BDTD, sistema AGUIA da Agência USP de Gestão da Informação Acadêmica, Books. Google, Scrum.org e Scrum Alliance.

A proposta / desenvolvimento do projeto se iniciou pela extração de um *corpus* de texto, e normalização dos dados utilizando a ferramenta *Orange 3*, extraindo dados de artigos do *The Guardian* do último mês com assunto relacionado a guerra da Ucrânia onde serão classificados por: óleo, gás, preço dos alimentos, inflação, usina nuclear, preço do trigo, economia global entre outras, que são assuntos de impacto global atualmente.

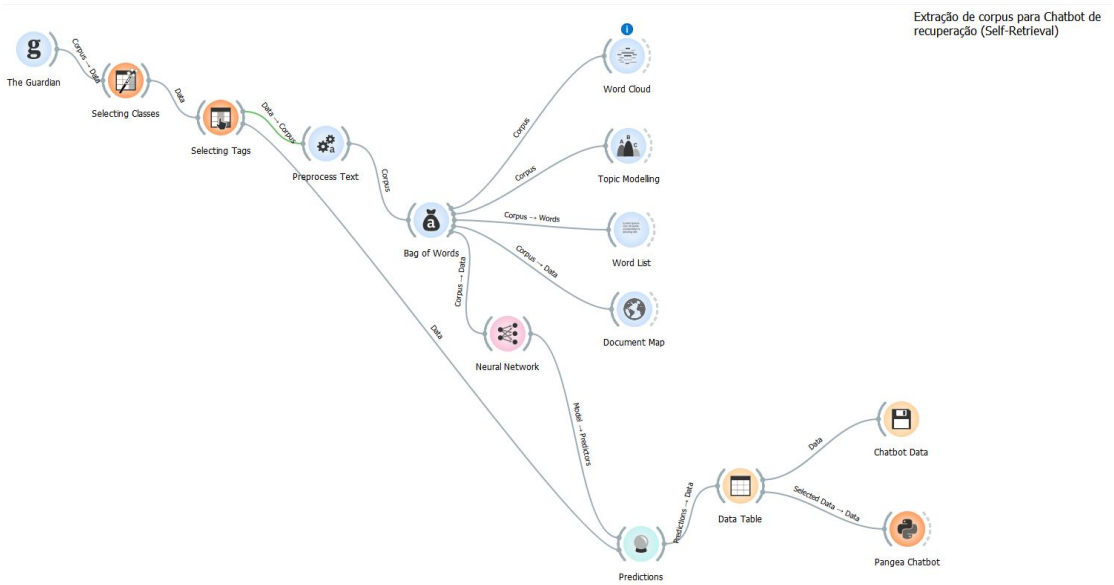


Figura 1: Orange 3 *Chatbot Canvas*

A seleção dos dados desejados ocorre no primeiro *widget* chamado *The Guardian* no canto superior esquerdo, selecionando a informação e os dados desejados. (Figura 2)

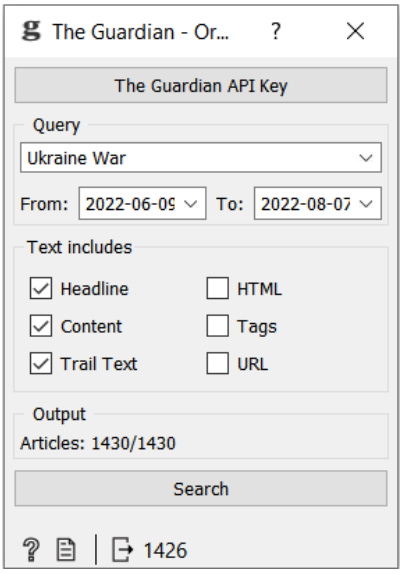


Figura 2 – *The Guardian* - Seleção dos dados

Em *query* foi inserido o item Guerra na Ucrânia (*Ukraine War)* com a seleção do último mês (junho / 2022) resultados retornados foram 1426 artigos selecionando apenas os títulos (*Headline*) e conteúdo (*Content*).

Após da extração dos dados efetuaremos a classificação das informações para facilitar a formatação do *corpus.*

Total de classes: 11

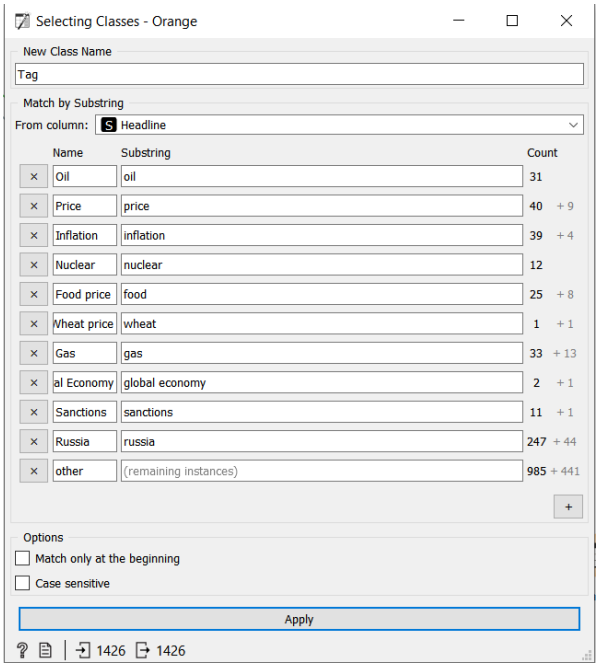


Figura 3 – Seleção das classes

Efetuada a seleção das classes que chamaremos de "*tag*", foi selecionado quais dados da mineração serão utilizados, neste so foram coletados os *headlines* e o *train text* que são frases curtas sobre o artigo encontrado.

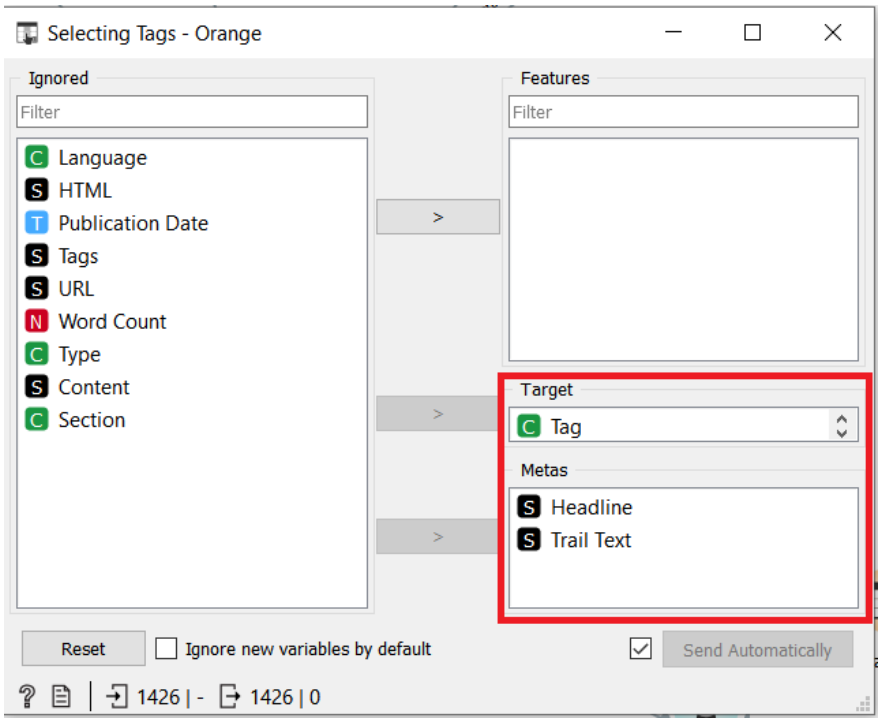


Figura 4 – Selecionando as *tags*

Após a extração dos dados foi efetuado o pré-processamento do *corpus* removendo *stopwords, urls, parse htm e* conversão das palavras em letras minúsculas. (*Figura 5)*

Tokenização e léxicos foram efetuados durante o pré-processamento para melhoria da acurácia dos dados de leitura *do chatbot* de recuperação *(Retrieval-based)*.

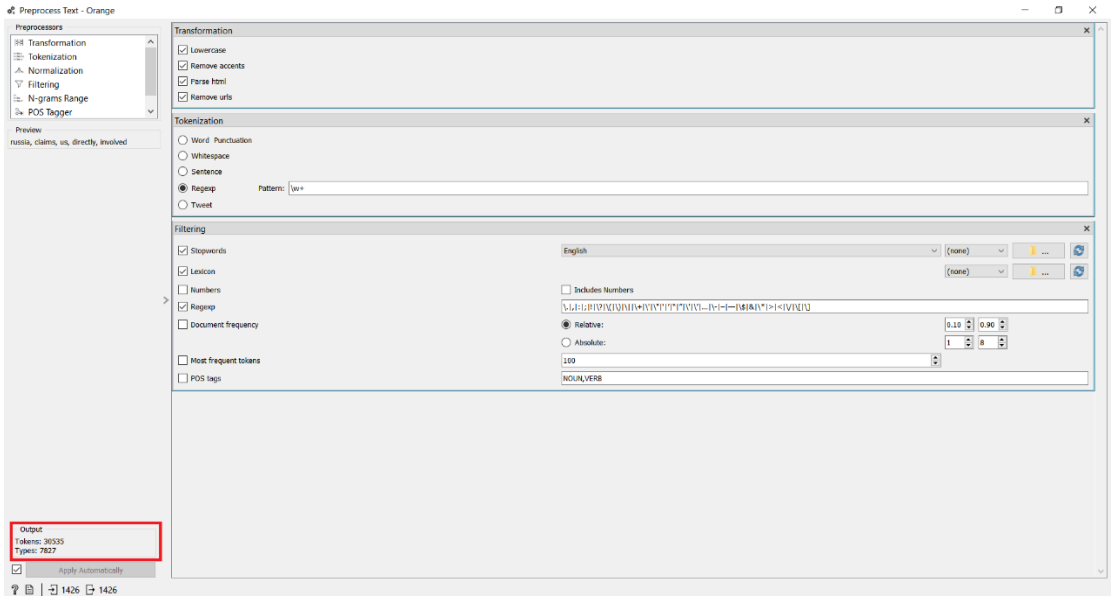


Figura 5 – Pré-processamento do texto (transformação e tokenização)

Quantidade de *tokens* identificados no pré-processamento: 30535

As informações serão filtradas sem os caracteres especiais \.|,|:|;|!|\?|\(|\)|\||\+|\'|\"|‘|’|“|”|\'|\’|…|\-|–|—|\$|&|*|>|<|\/|\[|\], \w. (Regexp) (Figura 6)

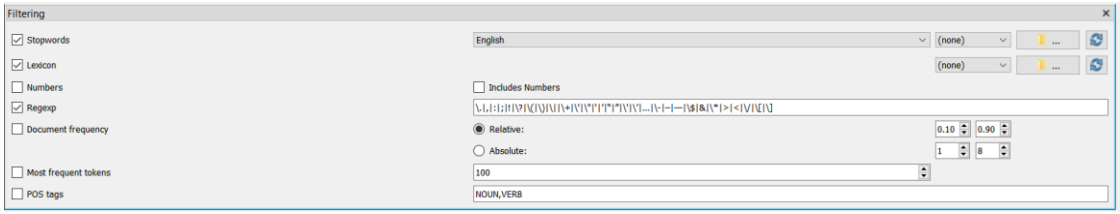


Figura 6 – Pré-processamento (*Filtering*)

Utilizando o método *Bag of Words* o algoritmo soma a quantidade de termos mais frequentes, porém com vetorização IDF (*IDF – Inverse document frequency*) que é a combinação de dois conceitos TF (*Termo of frequency*) e DF (*Document frequency*) que visa reduzir o peso de um termo se as ocorrências do termo estiverem espalhadas por todos os documentos, ajudando a normalizar os dados.

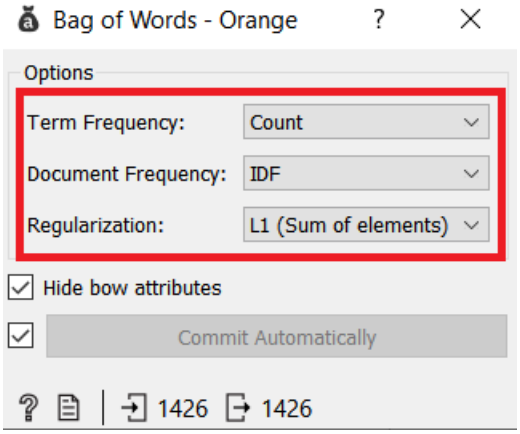


Figura 7 – *Bag of Words*

Após a aplicação do *Bag of Words* podemos realizar algumas análises antes do início do *chatbot*, como por exemplo saber as palavras de maior relevância para o tema utilizando o *widget Word Cloud*. Conforme figura abaixo, as palavras no centro são as que ocorrem com maior frequência e de dentro para fora gradativamente as que ocorrem com menor frequência.

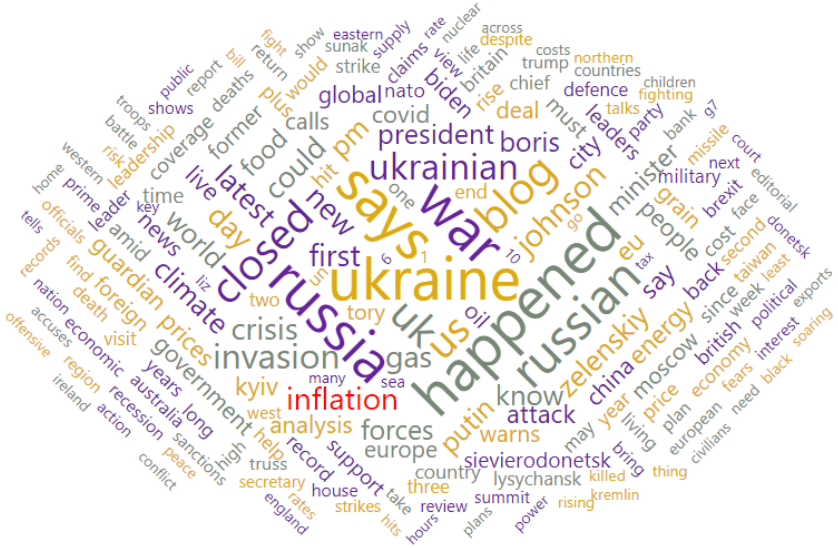


Figura 8 – *Word Cloud* (Preço do trigo – *Ukraine War*)

Analisando os dados por meio do *Word Cloud* podemos ver que devido ao conflito na Ucrânia as palavras que mais aparecem são *Russia, Inflation, War, Crisis. Etc.* que possuem impacto global conforme a frequência que as palavras aparecem.

Foi utilizado também o *widget Topic Modeling* para verificarmos as palavras chaves que aparecem durante a mineração destes dados, neste caso foram selecionados os 10 primeiros tópicos e podemos perceber algumas correlações com Ucrânia, Rússia, petróleo, combustível, inflação etc. (figura 9)

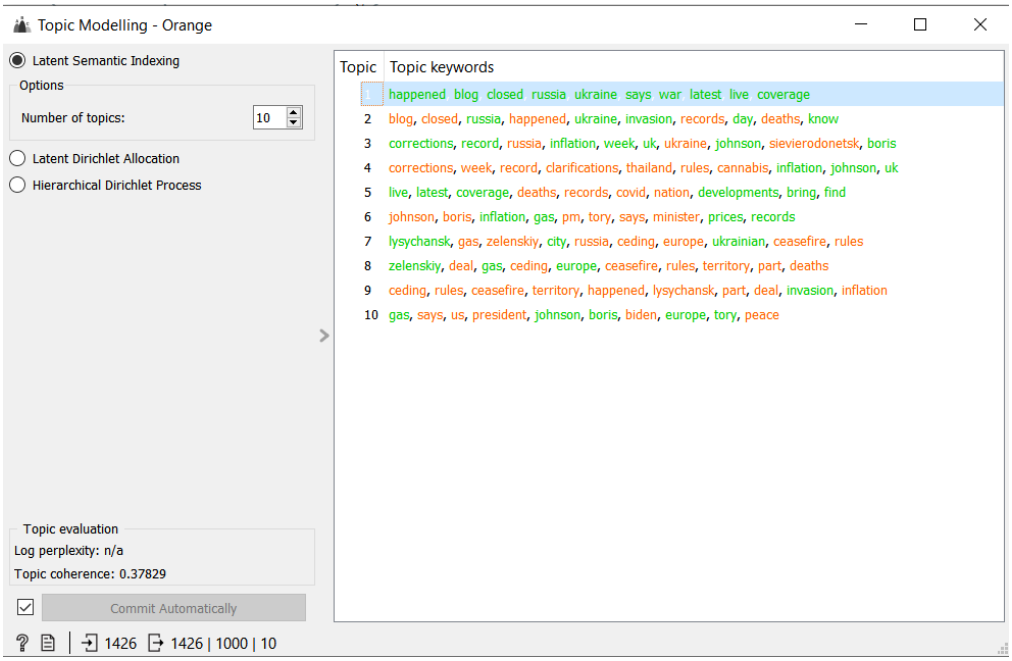


Figura 9 – *Topic Modeling* (Ukraine War)

Como uso do widget *Document Map* descobrimos quais países vem as notícias, neste caso as que possuem o tom vermelho mais forte são as que tem maior frequência, como exemplo a Rússia, Australia e China. (Figura 10)

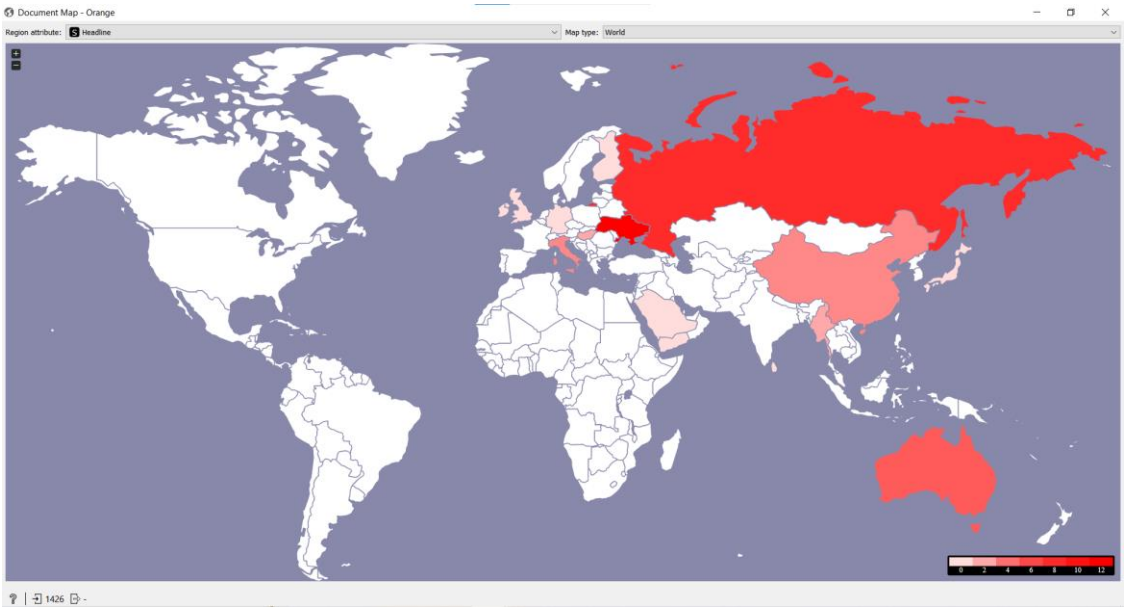


Figura 10 – *Document Map (Ukraine War*)

Redes Neurais utilizadas no *Chatbot*

As duas redes neurais utilizadas neste estudo são do tipo *Multi-Layer Perceptron (MLP)* da biblioteca *Sklearn* compostas por 3 camadas escondidas, essas utilizam um algoritmo de aprendizagem supervisionado que aprende uma função treinando em um conjunto de dados. Neste caso o conjunto alvo são os artigos de mineração relacionados ao conflito na Ucrânia, onde o treinamento pode aprender um aproximador de função não linear para classificação ou regressão, neste caso a função de ativação *Relu*. (figura 11)

O *solver* utilizado foi o SGD *(Stochastic Gradient Descent)* que é uma abordagem simples, mas muito eficiente para ajustar classificadores lineares e regressores sob funções de perda convexa, como máquinas vetoriais de suporte (linear) e regressão logística. O SGD é muito utilizado no contexto da aprendizagem em larga escala e um otimizador que auxilia no treinamento dos dados.

O objetivo deste primeiro conjunto de rede neural é separar os artigos provenientes da mineração e efetuar a correlação das 11 classes ou *"tags"* que foram mapeadas anteriormente: *oil, price, inflation, nucelar, food, wheat, gas, global economy, sanctions and Russia*, para o restante dos artigos que não forem similares a estas classificações estes serão separados na categoria *other*.

Por essa razão, o último *layer* ou camada dos neurônios no Orange 3 é **11** que é a quantidade de classes que foram mapeadas quando selecionamos os "*tags"* (categorias) para fazer a extração dos dados. Uma segunda rede neural de mesma configuração também está presente no *widget Python script* para treinamento do conjunto de dados e leitura do *Pangea Chatbot*.

A principal razão de utilização da função de ativação *Relu* é porque ele possui maior desempenho em dados esparsos, e o algoritmo do *Bag of Words* utiliza também dados esparsos, quanto mais neurônios existirem na camada mais esparsa será a representação, que neste caso é um conjunto de palavras que deverão ser lidas pelo *chatbot* antes de passarem pela segunda rede neural de treinamento (página 48)

Na prática, a ativação Relu (*Rectified Linear Unit*) tende a mostrar melhor desempenho de convergência do que a ativação Sigmóide, e neste primeiro conjunto está sendo utilizado para classificação de categorias por similaridade.

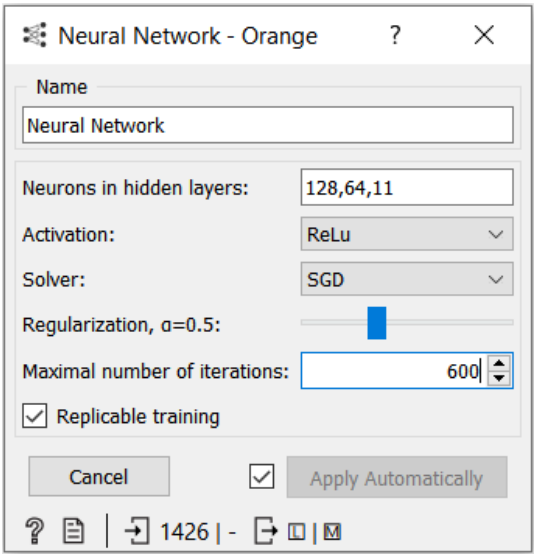


Figura 11 – Rede neural para processamento das categorias (tags) x similaridade de artigos

Após executar a rede neural com 600 iterações (épocas) foram efetuadas as predições para saber a similaridade dos artigos e as categorias, porém precisão dos dados ainda é baixa 0.47, uma segunda rede neural dentro *Python script* será executada para melhoria do conjunto de treinamento.

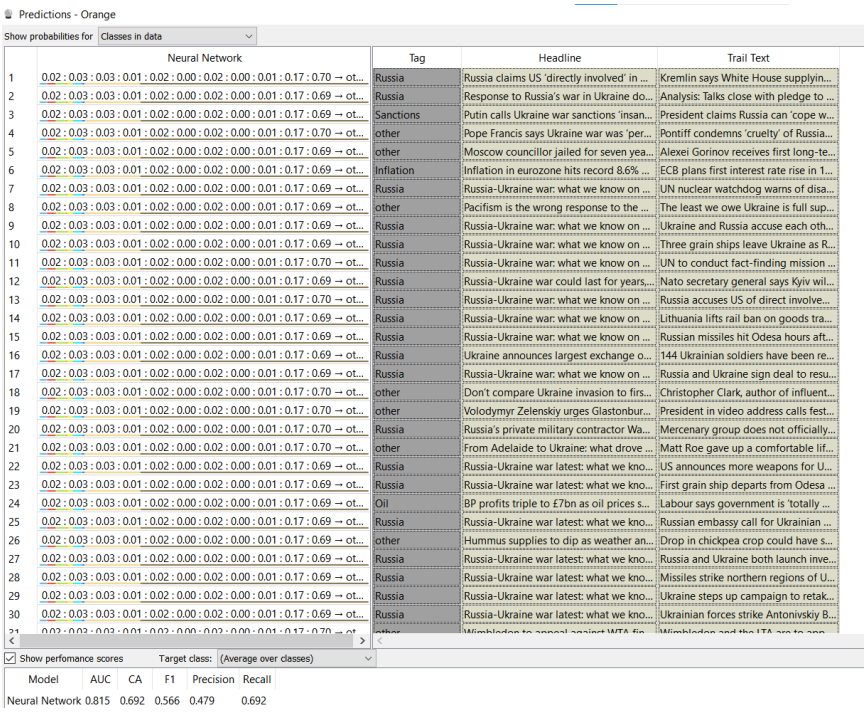
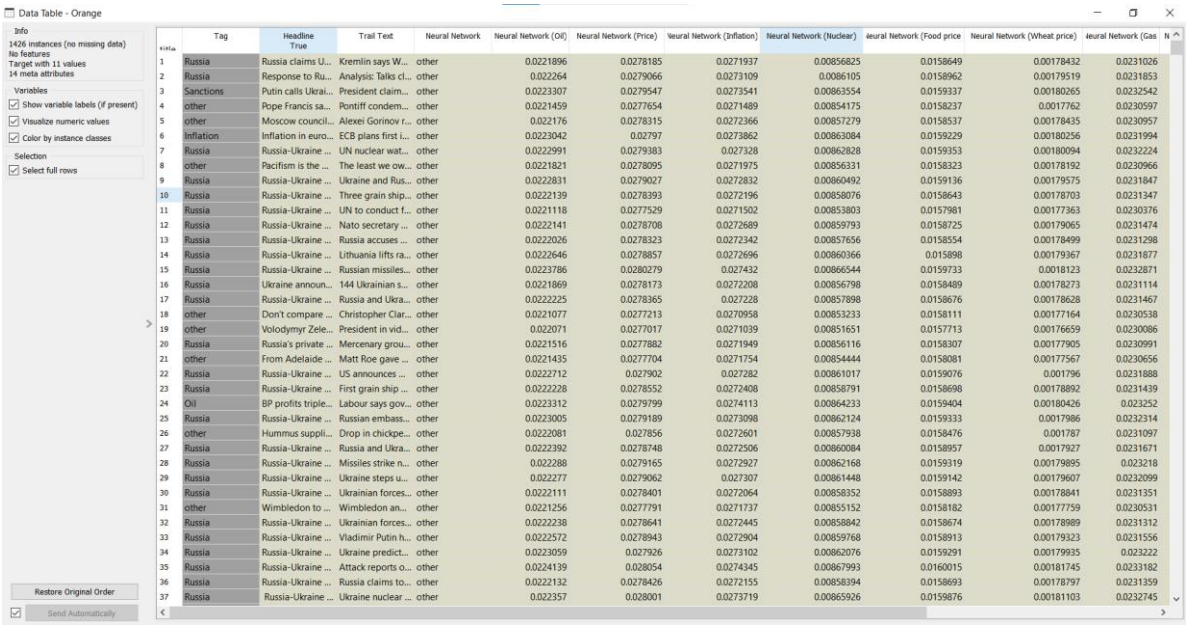


Figura 12 – predições para classificação dos dados

Após efetuar a predição dos dados, estes foram transferidos para uma tabela onde a informação será adicionada em um *corpus* formato. Json onde o *Pangea Chatbot* faz a leitura das respectivas perguntas e respostas.

Abaixo (figura 13) o resultado da similaridade entre as 11 categorias e os respectivos artigos extraídos do *The Guardian*.



	Tag	Headline True	Trail Text	Neural Network	Neural Network (Oil)	Neural Network (Price)	Neural Network (Inflation)	Neural Network (Nuclear)	Neural Network (Food price	Neural Network (Wheat price)	Neural Network (Gas
1	Russia	Russia claims U...	Kremlin says W...	other	0.0221896	0.0278185	0.0271937	0.00856825	0.0158649	0.00178432	0.0231026
2	Russia	Response to Ru...	Analysis: Talks cl...	other	0.022264	0.0279066	0.0273109	0.0086105	0.0158962	0.00179519	0.0231853
3	Sanctions	Putin calls Ukrai...	President claim...	other	0.0223307	0.0279547	0.0273541	0.00863554	0.0159337	0.00180265	0.0232542
4	other	Pope Francis sa...	Pontiff condem...	other	0.0221459	0.0277654	0.0271489	0.00854175	0.0158237	0.0017762	0.0230597
5	other	Moscow council...	Alexei Gorinov r...	other	0.022176	0.0278315	0.0272366	0.00857279	0.0158537	0.00178435	0.0230957
6	Inflation	Inflation in euro...	ECB plans first i...	other	0.0223042	0.02797	0.0273862	0.00863084	0.0159229	0.00180256	0.0231994
7	Russia	Russia-Ukraine ...	UN nuclear wat...	other	0.0222991	0.0279383	0.027328	0.00862828	0.0159353	0.00180094	0.0232224
8	other	Pacifism is the ...	The least we ow...	other	0.0221821	0.0278095	0.0271975	0.00856331	0.0158323	0.00178192	0.0230966
9	Russia	Russia-Ukraine ...	Ukraine and Rus...	other	0.0222831	0.0279027	0.0272832	0.00860492	0.0159136	0.00179575	0.0231847
10	Russia	Russia-Ukraine ...	Three grain ship...	other	0.0222139	0.0278393	0.0272196	0.00858076	0.0158643	0.00178703	0.0231347
11	Russia	Russia-Ukraine ...	UN to conduct f...	other	0.0221118	0.0277529	0.0271502	0.00853803	0.0157981	0.00177363	0.0230376
12	Russia	Russia-Ukraine ...	Nato secretary ...	other	0.0222141	0.0278708	0.0272689	0.00859793	0.0158725	0.00179065	0.0231474
13	Russia	Russia-Ukraine ...	Russia accuses ...	other	0.0222026	0.0278323	0.0272342	0.00857656	0.0158554	0.00178499	0.0231298
14	Russia	Russia-Ukraine ...	Lithuania lifts ra...	other	0.0222646	0.0278857	0.0272696	0.00860366	0.015898	0.00179367	0.0231877
15	Russia	Russia-Ukraine ...	Russian missiles...	other	0.0223786	0.0280279	0.027432	0.00866544	0.0159733	0.0018123	0.0232871
16	Russia	Ukraine announ...	144 Ukrainian s...	other	0.0221869	0.0278173	0.0272208	0.00856798	0.0158489	0.00178273	0.0231114
17	Russia	Russia-Ukraine ...	Russia and Ukra...	other	0.0222225	0.0278365	0.027228	0.00857898	0.0158676	0.00178628	0.0231467
18	other	Don't compare ...	Christopher Clar...	other	0.0221077	0.0277213	0.0270958	0.00853233	0.0158111	0.00177164	0.0230538
19	other	Volodymyr Zele...	President in vid...	other	0.022071	0.0277017	0.0271039	0.00851651	0.0157713	0.00176659	0.0230086
20	Russia	Russia's private ...	Mercenary grou...	other	0.0221516	0.0277882	0.0271949	0.00856116	0.0158307	0.00177905	0.0230991
21	other	From Adelaide ...	Matt Roe gave ...	other	0.0221435	0.0277704	0.0271754	0.00854444	0.0158081	0.00177567	0.0230656
22	Russia	Russia-Ukraine ...	US announces ...	other	0.0222712	0.027902	0.027282	0.00861017	0.0159076	0.001796	0.0231888
23	Russia	Russia-Ukraine ...	First grain ship ...	other	0.0222228	0.0278552	0.0272408	0.00858791	0.0158698	0.00178892	0.0231439
24	Oil	BP profits triple...	Labour says gov...	other	0.0223312	0.0279799	0.0274113	0.00864233	0.0159404	0.00180426	0.023252
25	Russia	Russia-Ukraine ...	Russian embass...	other	0.0223005	0.0279189	0.0273098	0.00862124	0.0159333	0.0017986	0.0232314
26	other	Hummus suppli...	Drop in chickpe...	other	0.0222081	0.027856	0.0272601	0.00857938	0.0158476	0.001787	0.0231097
27	Russia	Russia-Ukraine ...	Russia and Ukra...	other	0.0222392	0.0278748	0.0272506	0.00860084	0.0158957	0.0017927	0.0231671
28	Russia	Russia-Ukraine ...	Missiles strike n...	other	0.022288	0.0279165	0.0272927	0.00862168	0.0159319	0.00179895	0.023218
29	Russia	Russia-Ukraine ...	Ukraine steps u...	other	0.022277	0.0279062	0.027307	0.00861448	0.0159142	0.00179607	0.0232099
30	Russia	Russia-Ukraine ...	Ukrainian forces...	other	0.0222111	0.0278401	0.0272064	0.00858352	0.0158893	0.00178841	0.0231351
31	other	Wimbledon to ...	Wimbledon an...	other	0.0221256	0.0277791	0.0271737	0.00855152	0.0158182	0.00177759	0.0230531
32	Russia	Russia-Ukraine ...	Ukrainian forces...	other	0.0222238	0.0278641	0.0272445	0.00858842	0.0158674	0.00178989	0.0231312
33	Russia	Russia-Ukraine ...	Vladimir Putin h...	other	0.0222572	0.0278943	0.0272904	0.00859768	0.0158913	0.00179323	0.0231556
34	Russia	Russia-Ukraine ...	Ukraine predict...	other	0.0223059	0.027926	0.0273102	0.00862076	0.0159291	0.00179935	0.023222
35	Russia	Russia-Ukraine ...	Attack reports o...	other	0.0224139	0.028054	0.0274345	0.00867993	0.0160015	0.00181745	0.0233182
36	Russia	Russia-Ukraine ...	Russia claims to...	other	0.0222132	0.0278426	0.0272155	0.00858394	0.0158693	0.00178797	0.0231359
37	Russia	Russia-Ukraine ...	Ukraine nuclear ...	other	0.022357	0.028001	0.0273719	0.00865926	0.0159876	0.00181103	0.0232745

Figura 13 – tabela com os resultados das predições
(similaridade entre artigos x categorias)

Após o preparo do *corpus* com as informações, foi adicionado um *widget do Python Script* que possui o código de treinamento e execução do *chatbot* que consiste em:

Script 1 - Treinamento (Training Data)

- Treina utilizando o *corpus* (.json) com as informações resultantes da mineração do *The Guardian*
- Criação dos vetores para treinamento baseado nas classes “tags”.
- Pré-processamento dos dados (lemmanização, tokenização, stop words e etc.)
- Armazenamento de classes e palavras em dois arquivos extensão. Pkl

- Segundo conjunto de redes neurais:

  Utiliza rede MLP *(Multi-Layer Perceptron), Ativação Relu e SGD (Stochastic Gradient Descent)* explicados anteriormente no primeiro conjunto de redes neurais com a mesma quantidade de neurônios e camadas, porém a única diferença é a saída (terceira camada densa com 11 neurônios) utilizando a função de ativação *Softmax*.

  A função de ativação *Softwax* dimensiona números em probabilidades. A saída de um *Softmax* é um vetor (neste caso as palavras do Bag of Words) com probabilidades de cada resultado possível. As probabilidades neste vetor somam um para todos os resultados ou classes possíveis das *"tags"*.

  Quando ocorrer as requisições do usuário, o algoritmo do *Bag of Words* estará treinado com as possíveis probabilidades para prover as respectivas respostas. O treinamento destas probabilidades ocorreu com 800 iterações onde atinge o melhor nível de precisão, neste caso 1, informado na seção 4.1 testes e resultados.

  Arquitetura da rede neural:

  - Primeira camada 128 neurônios, relu, 0.5 *dropout* (entrada)
  - Segunda camada 64 neurônios, relu, 0.5 *dropout*
  - Terceira camada 11 neurônios, *softmax*, 0.5 *dropout* (neste caso o código lê a quantidade de "*tags*" ou classes no arquivo Json e modifica de acordo com a quantidade de categorias para melhor treinamento da rede neural (saída)

```
model = Sequential()
model.add(Dense(128, input_shape=(len(train_x[0]),), activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(64, activation='relu'))
model.add(Dropout(0.5))
model.add(Dense(len(train_y[0]), activation='softmax'))
```

Figura 14 – Rede neural 3 camadas (*Orange 3 – Python Script*)

Script 2 – *Pangea Chatbot*

- Abre os arquivos relacionados a classe, palavras e o *corpus;*
- Efetua novamente o pré-processamento dos dados (*tokeniza, lemaniza*, remove maiúsculas, minúsculas e caracteres especiais);
- Separa as palavras em um *array*;
- Executa o *Bag of Words*;
- Faz a predição entre as classes, perguntas e respostas;
- Abre a biblioteca gráfica no *Tkinter* para tela gráfica do *chatbot*;
- Inicia a conversa e vai computando o tempo médio de execução.

```
Running script:

1/1 [==============================] - ETA: 0s
1/1 [==============================] - 0s 104ms/step
```

Figura 15 – Medição de execução (Pangea Chatbot)

A tela gráfica é aberta durante a execução do *Pangea Chatbo*t onde o usuário pode fazer as perguntas relacionadas a crise na Ucrânia baseado nas categorias que buscamos na mineração: óleo, gás, preço dos alimentos, inflação, usina nuclear, preço do trigo, economia global.

A estrutura dos dados é estática e possibilita o *chatbot* achar as respostas rapidamente de acordo com a estrutura do Json, conforme exemplo abaixo:

```
{
   "tag":"nuclear",
   "patterns":[
      "How is the situation on the nuclear usine in Ukraine?",
      "Why countries are concerned about Ukraine nuclear plant?",
      "How nuclear disaster can impact the globe?"
   ],
   "responses":[
      "Russia-Ukraine war live: International Atomic Energy Agency
raises grave concerns over shelling at nuclear plant – as it happened",
      "Russia readies for southern offensive as alarm raised over
shelling of nuclear plant",
      "UN nuclear watchdog warns of disaster risk as Zaporizhzhia
power plant shelled; war about to enter a new phase, UK intelligence
predicts"
   ],
   "context":[
      ""
   ]
},
```

Figura 16 – Estrutura do arquivo Json (“tag” nuclear)

A chamada do *Pangea Chabot* ocorre dentro do *Orange 3* e abre graficamente a tela abaixo permitindo ao usuário fazer perguntas de acordo com as 11 categorias mencionadas nesse trabalho: *nuclear, russia. sanctions, wheat price, oil, gas, inflation e etc.*

As respostas ocorrem utilizando o algoritmo *Bag of Words (BOW)* que busca nos três arquivos auxiliares, a categoria (.pkl), *Bag of Words* de palavras (.pkl) e estrutura de perguntas e respostas (.JSON).

Note que as repostas serão compiladas utilizando o assunto específico do conflito da Ucrânia.

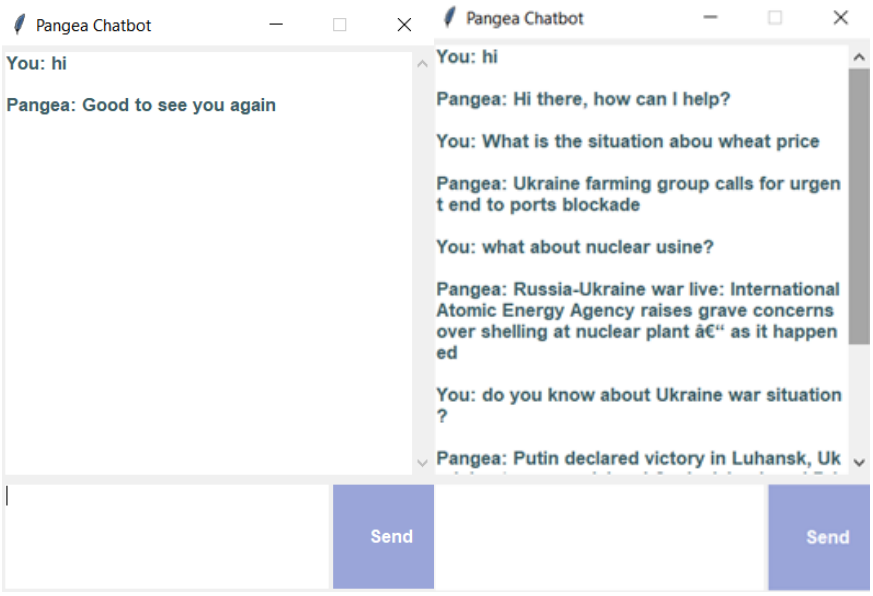


Figura 18 – Pangea Chatbot em execução

4.1 Testes e resultados

De acordo com os testes na rede neural utilizando 200, 600 e 800 iterações (épocas) o melhor nível de acerto foi utilizando 800 épocas com nível de precisão próxima de 1.

Por se um conjunto de dados estruturado e pequeno, as taxas de precisão são mais altas, porém quanto maior o número de dados no arquivo maior a complexidade de treino, mas para este trabalho selecionei uma rede neural com 800 iterações (épocas).

200 iterações (épocas) – Resultado dos testes

```
 1/11 [=>............................] - ETA: 0s - loss: 0.0016 - accuracy: 1.0000
11/11 [==============================] - 0s 3ms/step - loss: 0.0378 - accuracy: 0.9811
Epoch 200/200

 1/11 [=>............................] - ETA: 0s - loss: 0.3034 - accuracy: 1.0000
11/11 [==============================] - 0s 3ms/step - loss: 0.1232 - accuracy: 0.9434
model created and saved
>>>
```

Figura 14 – Treinamento 200 épocas

600 iterações (épocas) – Resultado dos testes

```
 1/11 [=>............................] - ETA: 0s - loss: 0.0017 - accuracy: 1.0000
11/11 [==============================] - 0s 3ms/step - loss: 0.0338 - accuracy: 0.9811
Epoch 600/600

 1/11 [=>............................] - ETA: 0s - loss: 1.3863e-04 - accuracy: 1.0000
11/11 [==============================] - 0s 3ms/step - loss: 0.0354 - accuracy: 0.9811
model created and saved
>>>
```

Figura 15 – Treinamento 600 épocas

800 iterações (épocas) – Resultado dos Testes

```
 1/11 [=>............................] - ETA: 0s - loss: 0.1418 - accuracy: 0.8000
11/11 [==============================] - 0s 3ms/step - loss: 0.0759 - accuracy: 0.9434
Epoch 800/800

 1/11 [=>............................] - ETA: 0s - loss: 0.0174 - accuracy: 1.0000
11/11 [==============================] - 0s 3ms/step - loss: 0.0246 - accuracy: 1.0000
model created and saved
>>>
```

Figura 16 – Treinamento 600 iterações (épocas)

Durante os testes também foi identificado que o *chatbot* tem dificuldade em respostas longas, ou seja, ele se perde quando é colocado um grande número de respostas ou frases longas, gerando erros e/ou ambiguidade nas respostas. Para solucionar este problema, cada categoria ou "*tag*" possui um número limitado de repostas de até 3 linhas e adicionei uma quantidade maior de perguntas para que o algoritmo consiga ter maiores possibilidades de achar as respostas, resumindo, o *corpus* deste *Chatbot* comete menos erros quando utiliza um maior número de perguntas e um menor número de repostas por categoria.

5 CONCLUSÃO

Foram pesquisados diversos algoritmos no desenvolvimento deste trabalho e foi identificado que o *Chatbot* que erra menos é aquele que utiliza perguntas e repostas curtas. Em outras bases de dados maiores e não estruturadas ele se perde nas respostas e precisa de um tratamento para estruturação dos dados para melhorar o desempenho do algoritmo.

A criação de um *corpus* confiável demora tempo e muitos testes têm de ser efetuados para que haja uma melhoria na qualidade e entendimento do algoritmo. Empresas se dedicam em criar bases de dados mais confiáveis para serem utilizadas no *Machine Learning* com times dedicados em aumentar semântica, reduzir ambiguidade etc. Para este trabalho o *corpus* criado é o mais simples possível e mesmo assim, tive que fazer diversos ajustes para conseguir as respostas considerando que o *Bag of Words* cria vetores grandes e esparsos.

As redes neurais ajudam no pré-processamento, mas também exigem poder computacional para conseguirem rodar a massa de dados, durante o desenvolvimento deste trabalho tive que reduzir quantidade de artigos de mineração, pois exigia muito poder computacional.

Quanto maior o pré-processamento dos dados, melhor a qualidade das respostas do *Chatbot.* Neste trabalho, o pré-processamento ocorre em três momentos, o primeiro é logo após a extração dos dados do "*The Guardian*" (*Pre Process Text*), depois na rede neural conectada (*Neural Network & Predictions*) que efetua as predições de similaridade das categorias ou "*tags*" e por último dentro do próprio *Python Script* utilizando uma segunda rede neural de mesma configuração informada durante esta pesquisa.

Integrações entre arquivos *json* e *ows* não ocorrem dentro do Orange 3, após o pré-processamento dos dados estes são inseridos sem integração dentro do *corpus* do *Chatbot.*Porém a chamada ocorre dentro do Orange 3 onde consegui instalar as bibliotecas do *NLTK*.

Durante a pesquisa também encontrei uma biblioteca de língua natural chamada *Chatterbot* porém esta já foi descontinuada desde a versão 3.6 do Python, esta biblioteca foi a minha primeira escolha para esta pesquisa, porém seu uso foi inviável.

Pesquisei outros algoritmos que também podem ser usados como *chatbot* buscando informações de um site Web por exemplo, no entanto, não há como efetuar as medições de treino de informação e geralmente ele se perde nas respostas.

Por fim, pesquisei também o *Document Embedding* que utiliza vetores mais densos. Porém a complexidade do algoritmo é maior para aprendizado e entendimento dos vetores a serem utilizados.

6 REFERÊNCIAS

Referências

[2] Eduardo Fagundes, Projetos de Inteligência Artificial. Link: Projetos de Inteligência Artificial - Eduardo Fagundes (efagundes.com) (Pag 4)

[4] Jowita Kessler, Artificial Intelligence implementation in 5 steps. Link: Artificial intelligence implementation in 5 steps - Neoteric

[6] Netcomm Learning, Artificial Intelligence for Project Managers. Link: How Artificial Intelligence is Impacting Project Management? | Project Management Tips - Bing video

[9] Anushka Agarwal**,** Cyfuture – How to create a chatbot using machine learning - How to Create a Chatbot using Machine Learning (cyfuture.com)

[10] 4 Fases de Um Projeto de Inteligência Artificial — Ciência e Dados (cienciaedados.com)

[18] Thiago Carvalo Dávila , 2018 - KINO: an approach for rule-based chatbot development, monitoring and evaluation (BDTD)
Description: KINO: an approach for rule-based chatbot development, monitoring and evaluation (ibict.br)

[19]Oberdan Alves de Almeida Junior, 2017 - Beck: um chatbot baseado na terapia cognitivo-comportamental para apoiar adolescentes com depressão (BDTD)
Description: Beck: um chatbot baseado na terapia cognitivo-comportamental para apoiar adolescentes com depressão (ibict.br)

[20] RODRIGO SOUZA WILKENS, 2016 - A study of the use of natural language processing for conversational agentes (BDTD)
Description: A study of the use of natural language processing for conversational agents (ibict.br)