

CHATBOT COM MINERAÇÃO DE EMOÇÕES

João Pedro Lacerda Pereira¹
Jaqueline Brigidori Pugliesi²

Resumo

Muitas empresas estão optando por utilizar os *chatbots* em alguns casos, como por exemplo, respondendo dúvidas no telemarketing. Sendo assim, as empresas estão explorando com muito empenho este mercado, que pode ser considerado recente. Conseqüentemente, é inegável a evolução dos *chatbots* e o índice de crescimento em relação a eles no mercado. Isso tudo gerou um grande número de tecnologias relacionadas ao assunto, entretanto, isto também mostrou que as empresas têm várias preocupações fundamentais que são: a eficiência com que seus clientes têm suas dúvidas respondidas, a forma com que eles são tratados e também o quanto eles podem economizar com esses dois fatores. Para que o cliente seja melhor atendido é interessante que os *chatbots* possam dar respostas com base nas emoções dos usuários, fazendo-os se sentirem confortáveis, mesmo em situações de estresse e proporcionando uma experiência mais agradável usando o serviço em específico. Essa preocupação gerou uma série de testes envolvendo ferramentas para o desenvolvimento de *chatbots* e mineração de emoções utilizando NLTK, os resultados se mostraram excelentes tanto com a biblioteca *chatterbot*, podendo ter uma conversa por voz, e a NLTK apresentando as emoções do usuário mesmo a base de dados inicialmente sendo pequena, os resultados obtidos foram muito bons.

Palavras-chave: *Chatbot*. Mineração de emoções. NLTK (*Natural Language Toolkit*).

Abstract

Many companies are opting to use chatbots in some cases, such as answering questions in telemarketing. Therefore, the companies are making a great effort to explore this market, that can be considered recent. Consequently, it is undeniable the chatbot's evolution, as well as their growth rate in the market. This has generated a large number of technologies related to this subject, however, this also showed that the companies have a number of major concerns: the efficiency in answering their customers' doubts, the way in which clients are treated and also and how much they can save with these two factors. In order to help the customer better, it is interesting that the chatbots can give answers based on the users' emotions, making them feel comfortable even in stressful situations and providing a more enjoyable experience using the specific service. This concern has generated a series of tests involving tools for the development of chatbots and the mining of emotions using NLTK. The results were excellent both with the chatterbot library, being able to have a voice chat, and with the NLTK, having very good results in showing the emotions of the user, even with an initially small database.

¹ Graduando em Análise e Desenvolvimento de sistemas pela Fatec Dr Thomaz Novelino – Franca/SP. Endereço eletrônico: joapedro.lacerda1@hotmail.com

² Doutora em Ciências da Computação pela USP – São Carlos/SP. Endereço eletrônico: jbpugliesi@gmail.com

Keywords: *Chatbot. Emotion mining. NLTK (Natural Language Toolkit).*

1 Introdução

O mercado de *chatbots* está sofrendo um aumento gigantesco, a consultoria gartner prevê que mais de 3 milhões de trabalhadores serão liderados por chefes robôs, e 20% dos cargos relacionados ao trabalho serão geridos por máquinas (EXAME, 2017).

Em outra pesquisa feita pela *Mindbowser*, é demonstrado que 75% das empresas têm, em seus planos, criar um *chatbot* ainda no ano de 2017. Já outro estudo feito pela *Juniper Research* demonstra que os *chatbots* irão gerar um grande aumento econômico a todas as empresas que estão utilizando os *chatbots* - apenas na área da saúde, que é atualmente uma das maiores usuárias de *chatbots*, o cálculo de redução de custo pode chegar a US\$ 3,6 Bilhões até 2022, dos quais serão poupados, ainda esse ano, US\$ 2,8 milhões (EXAME, 2017).

A principal questão que motivou este estudo é o quanto o conforto dos usuários é importante para as empresas a ponto de influenciá-las a criar um *chatbot* que dê respostas para o usuário com base na emoção que ele apresenta, visando ao seu conforto.

O objetivo deste trabalho é desenvolver um *chatbot* utilizando uma série de bibliotecas de Inteligência Artificial robustas, com mineração de emoções que pudesse dar respostas, levando em consideração as emoções apresentadas pelo usuário.

Inicialmente foi considerado utilizar o IBM Watson para fazer o desenvolvimento deste projeto, já que ele consegue identificar emoções, porém devido a algumas limitações de acessibilidade foi preciso levar em consideração outras opções, e a opção escolhida acabou sendo a biblioteca *chatterbot*, apesar da escassez de conteúdo na internet, fora a documentação e alguns poucos tutoriais mas sua documentação acabou sendo de grande ajuda por ser bem completa e amigável.

Com a grande massa de testes que foram feitos e com os resultados adquiridos é esperou-se alcançar um *chatbot* aplicado no telegram ou até mesmo em sua própria plataforma e também com uma base de dados bem maior que a base de dados utilizada.

Este artigo está estruturado de forma com que você possa ter um breve entendimento sobre a biblioteca *chatterbot*, ademais proporcionar e entender o

contexto e a evolução dos *chatbots*, saber um pouco sobre a biblioteca NLTK (*natural language toolkit*) e ter uma ideia em relação à mineração de textos e mineração de emoções. Em seguida, serão apresentados alguns dos resultados alcançados por todo este projeto.

2 Referencial teórico

Nesta seção será feita uma contextualização dos *chatbots* para a atualidade e o quanto eles estão crescendo no mercado.

2.1 História dos *chatbots*

Chatbot ou também robô de *chat* é um software que gerencia a troca de mensagens usando ou não *Machine Learning*. Ele é bem versátil no quesito implementação, podendo ser implementado em várias redes sociais de conversações populares, como por exemplo, *facebook*, *whatsapp*, *messenger*, *telegram* e etc. (GLOBALBOTS, 2018).

A ideia de *chatbot* começou em 1950 com Alan Turing que propôs uma definição de Inteligência Artificial conhecida como “teste de *Turing*” no qual: Um computador pode ser considerado “inteligente” se ele puder conversar com uma pessoa sem que a pessoa perceba que está conversando com um humano ou com um computador (PUSH, 2018).

Em 1966, *Joseph Weizenbaum*, um pesquisador do MIT (*Massachusetts Institute of Technology*), criou um software chamado Eliza, um robô de conversação que tinha como objetivo simular um psicólogo virtual. Ele aplicou em Eliza uma técnica de reformular frases dos usuários que falam com ela para parecer que o *chat* tinha um vocabulário muito extenso. Ela foi a primeira tentativa de desenvolvimento de um software que pudesse passar no teste de Turing, conseguindo reconhecer cerca de 250 tipos de frases, e é considerada a mãe dos *chatbots* (LANDSTEINERN, 2005).

Em 1972 foi desenvolvido o Parry, um robô de conversação. Enquanto Eliza era uma simulação de um terapeuta, Parry foi desenvolvida para simular um paciente com esquizofrenia, o contrário de Eliza, Parry tinha uma maneira de se comunicar de uma forma bem mais séria. Parry foi o primeiro robô de conversação a passar no teste

de *Turing*, pois os pesquisadores não conseguiam diferenciá-la de um indivíduo com esquizofrenia (LUN, 2017).

Em 1992 também foi desenvolvido o *Dr. Sabitso*, que era um programa com síntese de fala (um programa responsável por simular a fala humana) que foi feito para computadores pessoais, MSDOS (sistema operacional). Ele foi criado com o objetivo de ser algum tipo de psicólogo para o usuário, mas acabou tendo alguns problemas quando eram feitas perguntas que ele não entendia, resultando em respostas inapropriadas (CLASSICRELOAD, 2018).

Em 2006 veio o *IBM Watson*, que originalmente foi desenvolvido para competir em um programa de TV, onde ganhou de todos os participantes na época. Desde então a IBM (*International Business Machines*) anda demonstrando um grande investimento nele nas áreas de processamento de linguagem natural e *Machine Learning*, para extrair uma grande massa de dados, atualmente ele está sendo utilizado para análise de dados, atendimento aos clientes e reconhecimento visual (CERTYS, 2018).

2.2 *Chatbots* no seu negócio

O fato de se fazer a inclusão de um *chatbot* em um e-commerce é algo de extrema vantagem, pois este serviço tem como capacidade a disponibilidade para trabalho 24/7 (24 horas, 7 dias por semana), poder responder dúvidas mais comuns dos clientes e tornar uma experiência mais personalizável e dinâmica. Após um diálogo satisfatório com o *chatbot*, o cliente irá desejar usar o mesmo tipo de serviço quando houver outras ocasionais dúvidas.

Os *chatbots* podem ser também um grande avanço na área de pesquisa, otimizando, de maneira incrível, as pesquisas de opinião e satisfação, podendo até transformá-las em algo amigável.

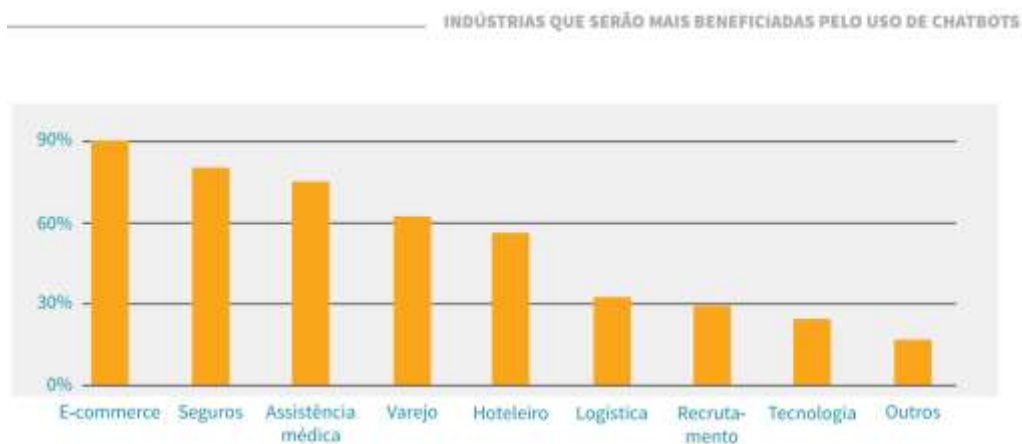
Os *chatbots* podem ser utilizados como até mesmo na área do entretenimento, podendo ter diversas aplicações, como por exemplo, se apresentarem como personagens virtuais para um jogo. Sendo capazes de conversar com internautas sendo uma fonte de entretenimento, vários desses aplicativos ficaram famosos, por suas respostas engraçadas e até mesmo alguns por adivinhar em que personagem fictício o usuário está pensando (GLOBALBOTS, 2018).

Apesar disso os *chatbots* podem ser muito importantes para a área da educação também, tendo *chats* que podem conversar com os alunos sobre suas respectivas dúvidas, ou até mesmo *chats* que podem conversar e ajudar os alunos a melhorar sua pronúncia em inglês (GLOBALBOTS, 2018).

Os *chatbot* podem ser muito úteis para manter um visitante de um respectivo site, gerando *leads* (Informações dos clientes ou usuários) para as equipes comerciais, podendo captar informações para integrar com ferramentas de força de vendas além de disparar e-mails e SMS (GLOBALAD, 2017).

Foi feita uma pesquisa conduzida pela *MindBownser*, a pesquisa é chamada de *chatbot survey* feita em 2017 que tem como objetivo principal, mostrar a tendência de mercado de *chabots* para os próximos anos, as Figuras 1 e 2 irão demonstrar as pesquisas mais relevantes em relação ao que foi dito nesta seção (TAKEBLOG, 2018).

Figura 1 – Industrias mais beneficiadas pelos *chabots*



Fonte: CHATBOTSURVEY, 2017, online.

Figura 2 – Funções de negócios que serão beneficiadas



Fonte: CHATBOTSURVEY, 2017, online.

3 Tecnologias e estudo de desenvolvimento

Nesta seção será abordada tecnologias e alguns conceitos importantes para o desenvolvimento deste trabalho.

3.1 Introdução a Biblioteca

Chatterbot é uma *Machine Learning* (aprendizado de máquina) que foi projetado com base em uma *engine* de conversações desenvolvida em Python que possibilita gerar respostas baseadas em conversas pré-armazenadas (CHATTERBOT, 2017).

Um diferencial da biblioteca *chatterbot* é a maneira com que ela foi desenvolvida, que permite que o *chatbot* possa aprender em qualquer idioma. Além disso, o *Machine Learning* que a biblioteca *Chatterbot* utiliza permite que ele melhore seu conhecimento conforme ele vai conversando com os usuários, claro, isso é opcional.

3.2 Treinamento

Quando o *chatbot* não é treinado, ele começa sem nenhum conhecimento de conversação (apenas quando é pré-programado com conversas), toda vez que um usuário envia uma declaração para o *chatbot*, e a biblioteca salva essa declaração, em seguida uma resposta é dada com base nessa declaração. Conforme o *chatbot* recebe mais informações, o tamanho do repertório de respostas dele aumenta junto com a precisão de cada resposta (CHATTERBOT, 2017).

A biblioteca *Chatterbot* inclui diversas ferramentas que simplificam o processo de treinamento do *chatbot*. O processo de treinamento que a biblioteca utiliza é a inclusão do carregamento de exemplo de diálogos (treinamento) no banco de dados do *chatbot*. Isso irá construir uma estrutura de dados que representa o conjunto de respostas que o *chatbot* poderá utilizar. Quando um treinamento (entrada com um conjunto de dados) é oferecido ao *chatbot*, ele cria entradas na estrutura de dados de conhecimento para que possibilite com que ele entenda as declarações de maneira mais específica e forneça respostas de uma forma mais precisa (CHATTERBOT, 2017).

Uma alternativa para o treinamento do *chatbot*, caso o usuário não queira mesmo treinar (alimentar a base de dados), é usando corpos de diálogo. O próprio *chatterbot* tem seu corpo de diálogo, além do mais, existem os corpos de diálogo do Twitter e do *Ubuntu*.

O corpo de diálogo do *Twitter* permite com que você treine o seu *chatbot* a partir dos dados que foram coletados da rede social (CHATTERBOT, 2017).

De acordo com Lowe et al. (2016), o corpo de diálogo do *Ubuntu* é um conjunto de dados contendo quase 1 milhão de diálogos, 7 milhões de expressões e 100 milhões de palavras.

O corpo de diálogo do *Ubuntu*, por ele ter uma grande quantidade de conjunto de dados, ele pode apresentar uma significativa perda de desempenho no treinamento do *chatbot*, e isto pode acarretar em um atraso no momento de gerar uma resposta, o que é algo para se levar em consideração caso resolva utilizá-lo. (CHATTERBOT, 2017).

De acordo com Weizeng (2018), o corpo de diálogo do *Twitter* é uma coleção de 12.696 de *tweets* e 4.232 trechos de conversa do *Twitter*. Isto permite que você treine seu *chatbot* com essa massiva base de dados.

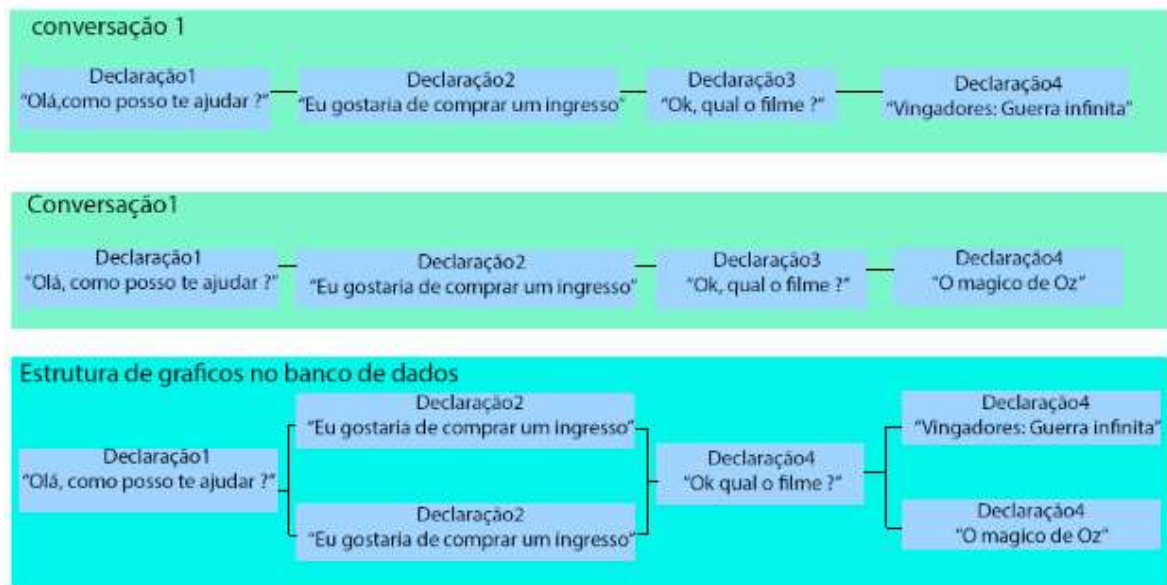
A biblioteca *chatterbot* também oferece seu próprio corpo de diálogo, que demonstrou ser um pouco mais modesto em comparação aos do *twitter* e do *Ubuntu* (CHATTERBOT, 2017).

3.3 Conversações

A biblioteca *chatterbot* permite múltiplas conversas simultaneamente, ou seja, ela suporta que múltiplos usuários falem com o *chatbot* ao mesmo tempo. Uma conversa é basicamente quando o usuário dá uma entrada com os dados e o *chatbot* responde com uma saída, o suporte á conversas simultâneas significa que o *chatbot* pode ter várias conversas diferentes com uma quantidade de pessoas diferentes (CHATTERBOT, 2017).

A biblioteca armazena conhecimento de conversas como declarações, cada declaração pode ter qualquer número de respostas possível como demonstrado na Figura 3 (CHATTERBOT, 2017).

Figura 3 - Diagrama de múltiplas conversações



Fonte: adaptado de CHATTERBOT, 2017, online.

Existe uma função no *chatbot* que é de interesse para quem quer fazer uma análise de dados em relação ao *chatbot* e ao o que o cliente conversa como, por exemplo, perguntas comuns, palavras chave, etc. Essa declaração se chama *ocorrencia* e indica o número de vezes que uma declaração em específico foi dada como resposta, isso ajuda o *chatbot* a determinar se uma resposta é mais usada que outra (CHATTERBOT, 2017).

3.4 Tipos de aprendizado

A biblioteca usa diferentes técnicas de *Machine Learning* para gerar as respostas, e os algoritmos em específicos vão depender da forma que o *chatbot* é usado e também das configurações dele (CHATTERBOT, 2017).

O algoritmo de pesquisa é uma parte importante de como o *chatbot* vai recuperar as possíveis declarações que ele pode responder de uma maneira mais eficiente e dinâmica. O algoritmo de pesquisa trabalha com uma forma mais rudimentar de Inteligência Artificial (CHATTERBOT, 2017).

Alguns exemplos de atributos que ajudam o *chatbot* com suas respostas são (CHATTERBOT, 2017):

- O quão parecido uma declaração de entrada é para com a declaração de saída.

- O quão frequentemente corre respostas conhecidas e semelhantes uma das outras.
- A probabilidade de uma declaração de entrada se encaixar em uma categoria na qual as declarações conhecidas fazem parte.

3.5 Algoritmo de classificação

O *chatbot*, dependendo de suas configurações e usos, tem uma vasta coleção de algoritmos de *Machine learning* para gerar suas respostas. (CHATTERBOT, 2017).

Um dos algoritmos mais importantes para gerar respostas é o teorema de Bayes, ele é uma fórmula matemática utilizada para calcular probabilidade, levando em conta de um evento que já aconteceu, também chamado de probabilidade condicional. O grande ponto do teorema de Bayes é que ele precisa de uma informação anterior, ou seja, ele precisa saber se um determinado evento já ocorreu e qual será a probabilidade deste mesmo evento ocorrer novamente (GONÇALVES, 2018).

Atualmente o algoritmo de Bayes se tornou popular na área de *Machine Learning* para categorizar os textos que são baseados na frequência das palavras usadas, e assim pode ser usado para identificar se determinado e-mail é spam ou sobre que assunto ele se refere (CANTIAGO, 2017).

A principal característica desse algoritmo é que ele é simples e rápido, e conseqüentemente seu desempenho é relativamente maior em relação aos outros classificadores. Outra qualidade do algoritmo de Bayes é que ele precisa de um número menor de dados para fazer classificações com uma boa precisão (CANTIAGO, 2017).

Muitos adaptadores lógicos da Biblioteca *Chatbot* usam algoritmos de classificação bayesiana ingênuos para determinar se uma instrução atende a um conjunto de critérios particulares que garantem que uma resposta seja gerada a partir destes critérios (CHATTERBOT, 2017).

3.6 NLTK

Processamento de linguagem natural é uma subárea da Inteligência Artificial que tem como objetivo uma compreensão das linguagens humanas, de forma que elas possam ser manipuladas por computadores. Existe uma série de bibliotecas para que facilite essa manipulação e essas bibliotecas são voltadas para os seguintes aspectos: fonologia, morfologia, sintaxe, semântica e pragmática (NLTK, 2009).

A NLTK ou *natural language toolkit* foi desenvolvida em 2001 como parte de uma disciplina de linguagem computacional no *Department of Computer and Information Science da University of Pennsylvania*. Desde então seu desenvolvimento tem crescido com a ajuda de dezenas de pessoas, graças ao software livre. Ela é uma biblioteca desenvolvida em *Python* para trabalhar com dados de linguagem natural, e oferece uma imensa quantidade de ferramentas de processamento de texto para classificação, tokenização, *streaming*, e análise e raciocínio semântico (NLTK, 2009).

3.7 Mineração de textos e emoções

A mineração de textos é uma das muitas áreas da Inteligência Artificial que tem como objetivo procurar padrões e conhecimento de interesse em textos. Grandes exemplos são o próprio mecanismo de pesquisa mais famoso do mundo, o *Google*, e os serviços de filtros de spam, que verificam se o e-mail é um spam ou não. Outro exemplo são os mecanismos de classificação de notícias que são usados para classificar que tipo de notícia se trata (GRANATYR, 2017).

De acordo com Gomes (2013 *apud* Adail, 2014, p.13), com o rápido crescimento da internet e a superpopulação das redes sociais o número de pessoas expressando suas opiniões sobre política, catástrofes, cinema e até mesmo serviços online ou offline. O que é interessante disso tudo é a maneira rápida que essas opiniões vão se propagando para outros usuários, fazendo com que eles também postem suas opiniões sobre tais assuntos. Essas opiniões podem ser tanto positivas quanto negativas.

Essa grande popularização da internet gera uma imensa quantidade de informações, e as organizações, em geral, por sua vez, não conseguem acompanhar essa grande massa de informação. No entanto, percebeu-se que ao analisar essas

informações dessa imensa massa de dados, eles poderiam conhecer as opiniões de seus usuários ou compradores sobre seus respectivos produtos e serviços.

As emoções são um assunto muito pesquisado em diferentes áreas como na psicologia e outras áreas responsáveis pelo estudo do comportamento, isso se deve ao fato de que a emoção é algo extremamente importante tanto para a conduta humana quanto para a natureza. Às emoções são uma linguagem universal (MARTINAZZO, 2010).

A análise de sentimentos é feita através de 3 passos:

1. Coleta de conteúdo: visa averiguar na web conteúdos sobre itens de interesse de quem está buscando, identificar se o que foi encontrado é relativo a um fato ou opinião, se for um fato ele é descartado pois o que está sendo buscado é uma opinião.
2. Classificação: identifica a popularidade do conteúdo. Polaridades são positivas, negativas ou neutras.
3. Sumarização dos resultados: visa tornar as classificações de diversas opiniões com fácil entendimento para o usuário, transformando de forma com que possa ser interpretado por um gráfico ou texto (GOMES, 2010).

4 Desenvolvimento e testes

Durante esta seção será abordado uma série de testes e conceitos relacionados ao desenvolvimento de um *chatbot* com mineração de emoções.

4.1 Desenvolvimento

No início do projeto optou-se por usar um ambiente de desenvolvimento *virtual envs* (máquinas virtuais) para que não houvesse quaisquer problemas com as versões do *Python* na hora de tratar com as bibliotecas, e nem erros de linguagem, e também para evitar quaisquer acidentes que pudesse ocorrer, e prevenir danos no sistema operacional utilizado.

Inicialmente, com o *chatbot*, foram realizados alguns testes de múltiplas conversações para verificar se realmente a biblioteca conseguia conversar com múltiplos clientes ao mesmo tempo.

Como mostra a Figura 4, foi desenvolvido um *chatbot* que não teve muito treinamento, utilizando 4 *prompts* (*interpretador de comandos de texto*) de comando diferentes para que simulasse 4 conversas simultâneas com o intuito de testar se a biblioteca poderia suportar múltiplos usuários conversando com o chat ao mesmo tempo.

Figura 4 - Teste de múltiplas conversações.

The image shows four terminal windows arranged in a 2x2 grid, each running a Python script named 'chats.py'. Each window displays a 'List Trainer' progress bar at 100%. The interactions are as follows:

- Top-left window:** User input: 'oi'; Chatbot response: 'como vai você?'; User input: empty.
- Top-right window:** User input: 'tudo bem?'; Chatbot response: 'estou bem e voce?'; User input: empty.
- Bottom-left window:** User input: 'olá'; Chatbot response: 'oi'; User input: empty.
- Bottom-right window:** User input: 'como vai você?'; Chatbot response: 'vou andando...'; User input: empty.

Fonte: Autor.

Em seguida, foram feitos alguns testes com o *speech recognition*, que é o responsável pelo reconhecimento de voz do *chatbot*.

Inicialmente, ocorreram muitos problemas para conseguir instalar algumas dependências do *speech recognition*, como o *pyaudio*, que é um driver extremamente necessário para quem vai usar a entrada de microfone. O problema foi resolvido usando uma distribuição do *Linux*, Deepin que é baseado no Debian. como mostrado na Figura 5, o reconhecimento de fala funcionou bem.

Figura 5 - Teste com o reconhecimento de voz.

```

joao@joao-PC: ~/Documents
joao@joao-PC:~/Documents$ python3 speechteste.py
ALSA lib pcm_dsnoop.c:638:(snd_pcm_dsnoop_open) unable to open slave
ALSA lib pcm_dmix.c:1099:(snd_pcm_dmix_open) unable to open slave
ALSA lib pcm.c:2565:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.rear
ALSA lib pcm.c:2565:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.center_lfe
ALSA lib pcm.c:2565:(snd_pcm_open_noupdate) Unknown PCM cards.pcm.side
ALSA lib pcm_dmix.c:1099:(snd_pcm_dmix_open) unable to open slave
Cannot connect to server socket err = No such file or directory
Cannot connect to server request channel
jack server is not running or cannot be started
JackShmReadWritePtr::~JackShmReadWritePtr - Init not done for -1, skipping unlock
JackShmReadWritePtr::~JackShmReadWritePtr - Init not done for -1, skipping unlock
diga alguma coisa :)
Você disse: Oi
Você disse: como vai
Você disse: paralelepipedo

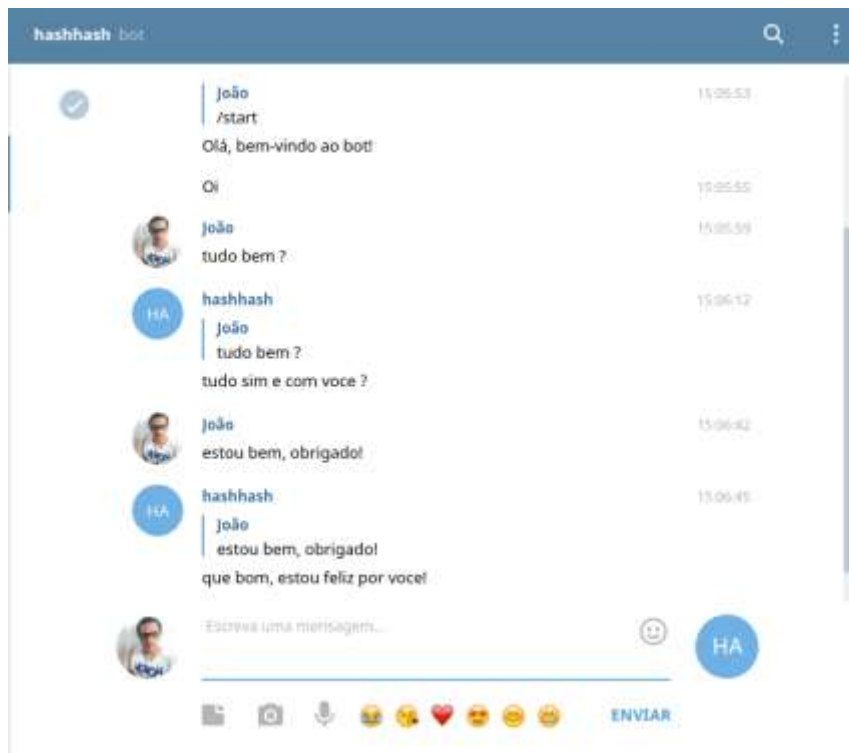
```

Fonte: Autor.

No começo, viu-se a necessidade de que o chat respondesse por voz, mas depois de alguns problemas relacionados a drivers e com a síntese de voz que apresentou não funcionar como esperado com uma voz alterada e de difícil entendimento. Julgou-se que não seria necessário, pelo menos inicialmente, que o *chatbot* respondesse por voz, e apenas permaneceu então que ele pudesse fazer o reconhecimento de voz.

Em seguida foi a hora de fazer o teste com o *chatbot* no *telegram*, já que ele oferece uma ferramenta que auxilia nessa implementação, tudo que você precisa é de um *token* que é utilizado para conectar o id do seu *chatbot* que foi criado lá no *telegram* com o *chatbot* que você desenvolveu por código. Como foi apenas um teste foi utilizado o corpo de diálogo da biblioteca *chatterbot* para não precisar treinar um *chatbot do zero*, também foi uma excelente oportunidade para *testar* o corpo de diálogo da biblioteca, como mostra a Figura 6. Os testes tiveram um resultado satisfatório.

Figura 6 - Testes de implementação no Telegram.



Fonte: Autor.

Em seguida depois do teste de implementação o *chatbot* no telegram serem satisfatório, teve início os testes com o reconhecimento de emoções utilizando a biblioteca NLTK.

Existe uma série de passos para a mineração de emoções em textos como, por exemplo, remoção das *stop-words* e extração do radical das palavras, além de mais uma série de abordagens de classificação para que possamos utilizar o Teorema de Bayes (incluso na biblioteca NLTK).

Stop-words, São palavras que são consideradas inúteis ou irrelevantes para a interpretação, como demonstrado na Tabela 1, e se elas estivessem lá, só gastariam tempo e processamento então é interessante fazer a sua remoção (MARCEL, 2009).

Tabela 1 – Exemplo de stop-words

E	Os	De
Um	Com	As
Do	Que	A

Fonte: Autor

Enquanto as *stop-words* são palavras consideradas inúteis, o radical ou também raiz é considerado muito importante para a interpretação, na Tabela 2 podemos ver um bom exemplo de radical, então é interessante fazer uma separação e armazenar (SÓPORTUGUÊS, 2017).

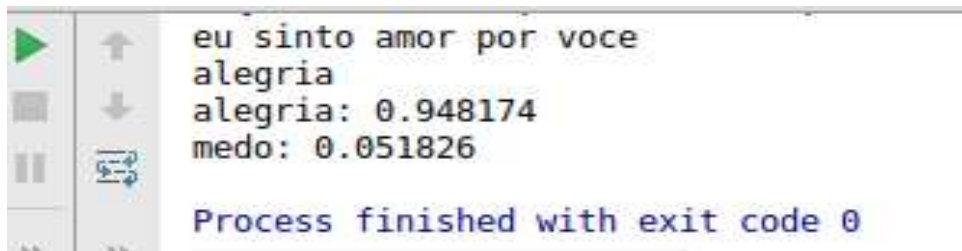
Tabela 2 – exemplo de radical

Cert-o
Cert-eza
In-cert-eza

Fonte: SÓPORTUGUÊS,2017, online.

Como podemos ver na Figura 7, o algoritmo classificou a frase: Eu sinto amor por você, como sendo uma frase de alegria, o que demonstra que o algoritmo está funcionando de acordo, porém após uma série de testes, ele demonstrou que não tem uma precisão tão grande em todas as frases que lhe são propostos.

Figura 7 - Primeiro teste envolvendo reconhecimento de emoções.



Fonte: Autor.

Como demonstrado na Figura 8, o algoritmo ainda não se mostra tão preciso quanto gostaria, cometendo alguns erros em algumas frases, que está relacionado à base de dados fraca até o momento. A Tabela 3 demonstra uma série de frases de alegria e medo, mostrando o quão preciso a biblioteca foi.

Figura 8 - segundo teste de reconhecimento de emoções

```

tenha um bom dia
alegria
alegria: 0.528678
medo: 0.471322

Process finished with exit code 0
    
```

Fonte: Autor.

Tabela 3 Frases e precisão da biblioteca

Frase	% alegria	% medo
A vida é boa	93 %	7 %
Estou muito feliz	62 %	38 %
O dia está lindo	80 %	20 %
Você é muito legal	82 %	18 %
Você tem um sorriso lindo	62 %	38 %
Essa casa me assusta	34 %	66 %
tenho medo de fantasmas	5 %	95 %
eu tomei um susto	13 %	87 %
eu estou encrencado	13 %	87 %
tome cuidado	3 %	97 %

Fonte: Autor

5 Resultados e discussão

Atualmente, com o crescimento dos *ChatBots* no mercado, nota-se que existe uma necessidade de *Chatbots* mais personalizáveis para um melhor desempenho de ferramenta utilizada pelos consumidores. Na primeira fase do projeto, foi feita uma pesquisa em relação aos *Chatbots* no mercado. Como esta tecnologia está crescendo, com o passar do tempo, nota-se a necessidade de uma biblioteca. Inicialmente, foi cogitado utilizar o IBM Watson para o projeto, por ele já ter uma API (*Application Programming Interface*) que reconhece as emoções. Porém, a principal ferramenta para este trabalho foi a *Chatterbot* por ter uma documentação muito amigável e mais completa.

Desde então, foram feitas pesquisas com a biblioteca em relação ao banco de dados e algumas maneiras de entradas de dados, sendo que a principal dificuldade em relação a ela foi entender como funciona o treinamento e também a forma com que ela aprende.

Assim, começaram-se alguns testes de conversação, de detecção de voz utilizando o *speech recognition* que demonstrou muitos problemas, inicialmente com o download dos drivers, mas funcionando eventualmente. Em seguida foi a tentativa de fazer o chat responder com voz utilizando o *pocket sphinx* mas optou-se por não deixar o chat com uma voz instável. Depois também se viu que não havia necessidade que o chat respondesse por voz em uma primeira versão. Em seguida foi feita toda uma pesquisa em relação a implementação do *chatbot* no *telegram*, e também em relação a mineração de emoções, optou-se por utilizar a biblioteca NLTK, apesar de ter tido alguns problemas em relação a base de dados que foi utilizada por ela ser uma base com apenas algumas frases, mas o resultado foi satisfatório, pois em vários casos o algoritmo dava respostas bem precisas levando em consideração sua base de dados.

Considerações finais

Os *chatbots* é uma área que está em constante crescimento, a quantidade de empresas que estão explorando este mercado, que é considerado novo, é muito alta. Isso demonstra que as empresas se preocupam tanto com o conforto do seu cliente quanto com os gastos necessários, sendo que uma boa alternativa é utilizar os *chatbots* que estão demonstrando ter um grande potencial. Tendo isso em vista foi pensado em realizar o desenvolvimento de um *chatbot* que não só se proporciona um baixo custo como também visa o conforto do cliente como uma prioridade, dando respostas com base nas suas emoções. No começo do desenvolvimento foi realizada uma extensa pesquisa em relações a linguagem e as ferramentas. Apesar das dificuldades encontradas, em relação a drivers e erros de compatibilidade, foi possível realizar todos os testes relacionados a implementação e a eficiência do *chatterbot* e da NLTK, assim como atingir o objetivo da implementação do reconhecimento de voz. A maioria dos testes que foram realizados com o *chatterbot* e com o *speech recognition* foram bem-sucedidos, e os testes com a implementação do telegram e também de mineração de emoções em Python foram bem satisfatórios, levando em considerando a base de dados do NLTK, foi possível adquirir respostas com alta precisão na identificação das emoções como por exemplo, felicidade e medo, o que é interessante levar para trabalhos futuros, fazendo um aumento da base de dados, fazer a implementação em outros serviços de *chat*, como o Messenger por exemplo

ou até mesmo fazendo o desenvolvimento de um aplicativo mobile para o *chatbot* utilizando a biblioteca *django*.

Referências

- CANTIAGO, **Algoritmo de classificação Naive Bayes**, SD, Disponível em: <https://www.organicadigital.com/seeds/algoritmo-de-classificacao-naive-bayes/>. Acessado em: 10. Out. 2018
- CERTSYS, **Infografico historia dos chabots**, SD, Disponível em: <https://www.certsys.com.br/insights/blog/infografico-historia-dos-chatbots/>. Acessado em: 25. set.2018
- CHATTERBOT, **Chatterbot**, SD, Disponível em: <https://chatterbot.readthedocs.io/en/stable/>. Acessado em: 20.Fev.2018
- CHATBOYSUVEY, **Status dos chatbots e perspectivas para 2017**, SD, Disponível em:< https://d335luupugsy2.cloudfront.net/cms/files/4333/1497910755Ebook_Chatbots_Survey_2017.pdf>. Acessado em: 15. set.2018
- CLASSICRELOAD, **DR-sabaitso**, SD, Disponível em <<https://classicreload.com/dr-sbaitso.html>>.Acessado em :15.Jul.2018
- DINO, **EXAME**, SD, Disponível em: < <https://exame.abril.com.br/negocios/dino/chatbots-movimentarao-mais-de-us-47-bilhoes-ate-2020-e-estao-na-mira-de-empresas-brasileiras/>>. Acessado em:19. Jun. 2018
- GLOBALBOTS, **Chatbot**, SD, Disponível em: <https://www.globalbot.com.br/chatbot>>Acessado em: 22.Set.2018
- GLOBALAD, **Os 7 principais usos de chatbots nas empresas**, SD, Disponível em: <http://globalad.com.br/blog/os-7-principais-usos-de-chatbots-nas-empresas/>. Acessado em 22. set.2018
- GOMESb ,**Analise de sentimentos**, SD, Disponível em:<<http://webholic.com.br/analise-de-sentimento/>> Acessado em:20.Jun.2018
- GOMES, 2013 *apud* ADAIL, 2014, p.13, **Mineração de textos : analise de sentimentos utilizando tweets da copa do mundo de 2014** ,SD, Disponível em:<<http://www.repositoriobib.ufc.br/000017/0000179f.pdf>>.Acessado em:20.Jun.2018
- GONÇALVES,2018 Teorema de Bayes, SD, Disponível em: < <https://www.voitto.com.br/blog/artigo/teorema-de-bayes>>. Acessado em: 10. out.2018
- GRANATY, **curso de mineração de emoções em textos em python**, SD, Disponível em: <<https://www.portalgsti.com.br/cursos/curso-mineracao-de-emocao-em-textos-com-python-e-nltk/>> Acessado em:21.Jun.2018
- INBOT, **História dos Chatbots**, SD, Disponível em : <https://www.inbot.com.br/chatbots/historia-dos-bots.php>> Acessado em:15.Mar.2018

LANDSTEINERN, **Elizabot**,SD, Disponível em: <<https://www.masswerk.at/elizabot/>>.
Acessado em 15. Mar.2018

LOWE, **Ubuntu dialogue corpus**, SD, Disponível em: < <https://arxiv.org/abs/1506.08909>>.
Acessado em:17. Mar.2018

LUN, **Parry**, SD, Disponível em <<https://www.chatbots.org/chatbot/parry/>> Acessado em:
16.Mar.2018

MARCEL, **stop words – como funcionam palavras de parada?**, SD, Disponível em:<
<https://www.agenciamestre.com/seo/stop-words-como-funcionam-palavras-de-parada/>>.
Acessado em:1. out.2018

MARTINAZZO, **Mineiração de emoções** ,SD, Disponível em:
<https://www.ppgia.pucpr.br/~paraiso/mineracaodeemocoas/recursos/barbara_martinazzo_v_ersaofinal.pdf>.Acessado em:1.jun.2018

NLTK, **NLTK BOOK**, SD, Disponível em: <<http://www.nltk.org/book/ch00-pt.html>>. Acessado
em:25.Jun.2018

NLTK, **Natural Language Toolkit**, SD, Disponível em: <<https://www.nltk.org/>>. Acessado
em:25. Jun. 2018

PHYLLIIDA, **Dialogue-Datasets**, SD, Disponível em:< <https://github.com/Phylliida/Dialogue-Datasets>>. Acessado em:16. Mar.2018

PUSH, **historia do chatbot:saiba como tudo começou**, SD, Disponível em:<
<https://push.al/historia-do-chatbot-saiba-como-tudo-comecou/>>. Acessado em 25.Set.2018

SÓPORTUGUÊS, **Raiz e radical de uma palavra**, SD, Disponível em :<
<https://www.soportugues.com.br/secoes/morf/morf2.php>>. Acessado em 1.Out.2018

TAKEBLOG, **Chatbot survey pesquisa global sobre chatbots**, SD, Disponível em:<
<https://take.net/blog/one-step-beyond/chatbot-survey-pesquisa-em-portugues/>>. Acessado
em: 5.Set.2018

WEIZENG, **Twitter dialogue Corpus**, SD, Disponível
em:<<https://github.com/hongweizeng/Dialogue-Corpus/tree/master/twitter>>. Acessado em:
17.Mar.2018