UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA

Jorge Henrique Cordeiro Linhares

Usain Bots: Uma proposta de construção de chatterbots por meio de um serviço web

---

Trabalho de Graduação

RECIFE, DEZEMBRO DE 2016

UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA

Jorge Henrique Cordeiro Linhares

Usain Bots: Uma proposta de construção de chatterbots por meio de um serviço web

Projeto de Graduação apresentado no Centro de Informática da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do grau de Bacharel em Sistemas de Informação.

Orientador: Vinicius Cardoso Garcia

RECIFE, DEZEMBRO DE 2016

UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA

Jorge Henrique Cordeiro Linhares

Usain Bots: Uma proposta de construção de chatterbots por meio de um serviço web

Projeto de Graduação apresentado no Centro de Informática da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do grau de Bacharel em Sistemas de Informação.

Orientador: Vinicius Cardoso Garcia

Aprovado em: ____/____/____.

BANCA EXAMINADORA

_____

Vinicius Cardoso Garcia
Universidade Federal de Pernambuco

_____

Kiev Santos da Gama
Universidade Federal de Pernambuco

RECIFE, DEZEMBRO DE 2016

Agradecimentos

Agradeço primeiramente a Deus em reconhecimento de que todas as coisas são Dele, para Ele e por Ele.

Agradeço a minha família por todo incentivo, amor, dedicação e paciência

Agradeço ao meu orientador Vinícius Garcia pela sua disponibilidade e por todo o apoio fornecido.

Agradeço a minha namorada Sandrine pela compreensão e por estar sempre presente para me ajudar.

Agradeço a todos os professores do CIn por todo o conhecimento e dedicação fornecidos durante o curso.

Agradeço aos meus amigos pelo incentivo e apoio durante toda a graduação.

Resumo

Atualmente os avanços tecnológicos estão engajando a geração, não só de novos meios tecnológicos, mas como também reanimando tecnologias que acabaram por ser inviabilizadas em seu surgimento. Seja devido a falta de recursos de *hardware* ou de processamento suficientes para suprir a sua existência. Isto ocorreu com a Inteligência artificial, oriunda dos anos 50-60, que ressurge ao mercado com grande poder de inovação para as empresas e com uma variedade vasta de subcampos, definidos em um escopo geral como mecanismos de aprendizagem e percepção. O processamento de linguagem natural (PLN), é um dos subcampos que em conjunto com a linguística, estuda os problemas da geração e compreensão automática de línguas humanas naturais. Seguindo esta linha de pesquisa, estão inseridos os *chatterbots (bots)*, que são ferramentas tecnológicas que se utilizam de processamento de linguagem natural para suprir uma necessidade de diálogo automatizado com os seres humanos. Este trabalho propõe a compreensão e estudo dos métodos de PLN para a construção de *bots*, assim como o desenvolvimento de um serviço *web* capaz de gerenciar *bots* e transmitir todas suas características e essências. A utilização do serviço será demonstrada de maneira prática com o CIn Bot, um *bot* que irá atender as demandas e necessidades dos alunos de graduação do Centro de Informática.

Palavras chave: *Chatterbots, processamento de linguagem natural, sistemas de recuperação da informação.*

Abstract

Currently, the technological advances are engaging not only generation of new technological means, but also reviving technology that turned out to be unfeasible in its emergence. Whether due to lack of hardware resources or enough processing to supply its existence. This occurred with the artificial intelligence, from 70's and 80's, which resurfaces to the market with great power of innovation for companies and with a vast variety of subfields, defined in a general scope as mechanisms of learning and perception. The natural language processing (NLP) is one of the subfields that, in conjunction with Linguistics, studies the problems of automatic generation and understanding of natural human languages. Following this line of research are inserted the chatterbots (*bots*), which are technological tools that use natural language processing to meeting for a need for automated dialogue with human beings. This paper proposes the understanding and study of the methods of PLN for construction of *bots*, as well as the development of a web service capable of managing bots and pass all its characteristics and essences. The use of the service will be demonstrated in a practical way with the CIn Bot, a bot that will meet the demands and needs of undergraduates of Center for Informatics.

Keywords: *Chatterbots, natural processing language, Information retrieval systems.*

Sumário

1. Introdução **15**

1.1 Motivação 15

1.2 Objetivos 15

1.2.1 Gerais 15

1.2.2 Específicos 16

1.3 Organização do trabalho 16

2. Fundamentação Teórica **17**

2.1 Inteligência Artificial 17

2.1.1 The Imitation Game 18

2.2 Chatterbots 19

2.2.1 Contexto histórico 19

2.2.2 Características e Classificações 20

2.2.3 Modelos de aquisição de conhecimento 21

2.3 Processamento de Linguagem Natural 22

2.3.1 Níveis de entendimento 22

2.3.2 Processamento 23

2.3.2.1 Aplicações 23

2.3.2.3 Estratégias 24

3. Usain Bots **27**

3.1 Proposta 27

3.1.1 Aplicação das características de um bot 27

3.1.2 Modelo de aquisição de conhecimento 28

3.2 Arquitetura 28

3.2.1 Node.js e Express 28

3.2.1.1 Arquitetura do Node.js 28

3.2.1.2 Arquitetura MVC com Node.js e Express 29

3.3 Usain Bots 31

3.3.1 Módulo de Comunicação 31

3.3.2 Módulo de Controle 31

3.3.2.1 Mecanismo de autenticação 32
3.3.2.2 Controladores 33
3.3.2.3 Modelos e Base de dados 33
3.3.3 Módulo de Processamento de Linguagem Natural 36
3.3.3.1 Extração 36
3.3.3.2 Classificador 36
3.3.3.3 Pesagem 37
3.3.4 Módulo de extração de dados 38
3.4 Considerações finais 39
4. Utilização do Usain Bots na construção do bot do CIn 40
4.1 Arquitetura 40
4.1.2 Controlador 40
4.1.3 View 41
4.2 Utilização do serviço 41
4.2.1 Identificação de abertura e cumprimento 42
4.2.2 Identificação de questionamento e proposição de sugestões 42
4.2.3 Identificação de questionamento e sem resposta viável 43
4.2.4 Identificação de agradecimento 43
4.2.5 Funções extras 44
4.2.6 Identificação de encerramento 45
5. Conclusão e trabalhos futuros 46

Lista de Tabelas

Tabela 1 - Algumas definições de inteligência artificial, organizadas em quatro categorias **17**

Tabela 2 - Categoria de bots **21**

Lista de Figuras

Figura 1 - O processo de recuperação da informação 24

Figura 2 - Orquestração de solicitação-resposta do Node.js 29

Figura 3 - Modelo de solicitação e resposta do Usain Bots 30

Figura 4 - Representação lógica da arquitetura do Usain Bots 31

Figura 5 - Arquivo auth.js e método isAuthenticated 32

Figura 6 - Mapeamento de rotas do controlador bot.js 33

Figura 7 - Esquema MongoDB da coleção de dados "Aplicação" 33

Figura 8 - Esquema MongoDB da coleção de dados "Bot" 34

Figura 9 - Representação visual de um "Documento" 34

Figura 10 - Esquema MongoDB da coleção de dados "Documento" 35

Figura 11 - Esquema MongoDB da coleção de dados "Evento" 35

Figura 12 - Esquema MongoDB da coleção de dados "Notícia" 36

Figura 13 - Processamento de Linguagem Natural 36

Figura 14 - Método getContentWiki do coletor geral 38

Figura 15 - Método getNews do coletor específico do CIn Bot 39

Figura 16 - Controlador app.js do CIn Bot 40

Figura 17 - Controlador Chat.js do CIn Bot 41

Figura 18 - View chat.html do CIn Bot 41

Figura 19 - CIn bot: Identificação de abertura e cumprimento 42

Figura 20 - CIn bot: Identificação de questionamento 42

Figura 21 - CIn bot: Ação de engajamento por sugestão 43

Figura 22 - CIn bot: Identificação de questionamento sem resposta viável 43

Figura 23 - CIn bot: Identificação de agradecimento 44

Figura 24 - CIn bot: Identificação de notícias e eventos 44

Figura 25 - CIn bot: Identificação de encerramento do diálogo 45

Capítulo 1

1. Introdução

Este capítulo apresentará a motivação necessária para a realização deste trabalho, objetivos a serem atingidos e a maneira ao qual foi estruturado, a fim de auxiliar um bom entendimento ao leitor.

1.1 Motivação

O avanços tecnológicos sobre a comunicação, informação e interação humana tem trazido consigo várias contribuições para a sociedade. Como resultado desses avanços, hoje temos agentes inteligentes[1] que sabem a rotina das pessoas, os seus gostos e desejos. Os esforços necessários para o desenvolvimento destas soluções acontecem em função da Inteligência Artificial, uma ciência que estuda métodos para desenvolver uma maior interação entre o ser humano e a máquina, da maneira mais natural possível. Uma destas maneiras, acontece por meio dos *chatterbots* que representam ferramentas de *software* que funcionam por meio de um terminal de diálogo, apresentando um comportamento conversacional humano em um determinado domínio de conhecimento. Entretanto, fazer a máquina compreender e propor um diálogo em linguagem natural não é uma tarefa simples. Para isso, estas ferramentas tecnológicas se utilizam de processamento de linguagem natural, uma subárea da inteligência artificial que tem se dedicado a analisar e promover a compreensão.

Levando em consideração a capacidade que essas ferramentas têm em interagir com o ser humano por meio de diálogo e pela sua capacidade de adaptação a novos domínios de conhecimento, é de viável a elaboração de um serviço inteiramente dedicado a construção destas ferramentas com a aplicação de técnicas bem estabelecidas de PLN, de maneira que se torne possível elaborar um modelo vigente de construção e gerenciamento de bots a partir de um serviço *web*.

1.2 Objetivos

1.2.1 Gerais

O objetivo deste trabalho é compreender e trabalhar conceitos inerentes à inteligência artificial, mais precisamente sobre o processamento de linguagem natural para a criação de um serviço *web* de criação e gerenciamento de *bots*.

[1] "Agente inteligente – Wikipédia, a enciclopédia livre." 2011. 22 Nov. 2016 <https://pt.wikipedia.org/wiki/Agente_inteligente>

1.2.2 Específicos

Este trabalho possui como objetivos específicos:

- Desenvolvimento de uma arquitetura de *software* que siga a proposta de um serviço *web* que atenda as características e essências dos *bots.*

- Analisar recursos dos serviço *web* proposto por este trabalho na aplicação cliente do Centro de Informática (CIn) da Universidade Federal de Pernambuco (UFPE).

- Aplicar técnicas e estratégias de Processamento de Linguagem Natural em um Sistema de Recuperação da Informação.

1.3 Organização do trabalho

O presente trabalho está organizado em 5 capítulos, dos quais o primeiro trata-se da introdução e os 4 demais podem ser descritos abaixo:

- Capítulo 2 apresenta um conjunto de definições sobre Processamento de Linguagem Natural, Inteligência Artificial e *Chatterbots*.

- Capítulo 3 apresenta o modelo de serviço *web* para *bots* proposto.

- Capítulo 4 apresenta a aplicação do modelo de serviço *web* para *bots* por meio do bot do CIn.

- Capítulo 5 são apresentados considerações finais do trabalho assim como as propostas para trabalhos futuros.

Capítulo 2

2. Fundamentação Teórica

Este capítulo abordará conhecimentos essenciais para a boa compreensão e continuidade deste trabalho. Propondo uma visão geral sobre os temas: *Chatterbots*, Inteligência Artificial, Processamento de Linguagem Natural e *Application Programming Interface*.

2.1 Inteligência Artificial

Desde sua origem, a inteligência artificial (artificial intelligence ou IA) nunca avançou tanto como ultimamente. Diversas empresas de tecnologia estão ampliando o seu arsenal de pesquisa e se apoiando na criação de novos recursos e serviços em nuvem que dão a inteligência artificial todo o poder de processamento e armazenamento que ela outrora não possuía. A definição de IA foi modelada com o passar dos anos. A tabela 1 representa 8 definições de inteligência artifical sobre os mais diversas perspectivas [15] onde a parte superior da tabela se relacionam a processos de pensamento e raciocínio, enquanto as definições da parte inferior se referem ao comportamento. As definições do lado esquerdo medem o sucesso em termos de fidelidade ao desempenho humano, enquanto as definições do lado direito medem o sucesso comparando-o a um conceito ideal de inteligência, chamado de racionalidade.

Pensando como um humano	Pensando racionalmente
“O novo e interessante esforço para fazer os computadores pensarem (...) máquinas com mentes, no sentido total e literal.” (Haugeland, 1985) “[Automatização de] atividades que associamos ao pensamento humano, atividades como a tomada de decisões, a resolução de problemas, o aprendizado...” (Bellman, 1978)	“O estudo das faculdades mentais pelo uso de modelos computacionais.” (Charniak e McDermott, 1985) “O estudo das computações que tornam possível perceber, raciocinar e agir.” (Winston, 1992)
Agindo como seres humanos	Agindo racionalmente
“A arte de criar máquinas que executam funções que exigem inteligência quando executadas por pessoas.” (Kurzweil, 1990) “O estudo de como os computadores podem fazer tarefas que hoje são melhor desempenhadas pelas pessoas.” (Rich and	“Inteligência Computacional é o estudo do projeto de agentes inteligentes.” (Poole et al., 1998) “AI... está relacionada a um desempenho inteligente de artefatos.” (Nilsson, 1998)

Knight, 1991)	

***Tabela 1** - Algumas definições de inteligência artificial, organizadas em quatro categorias [15].*

2.1.1 The Imitation Game

O levantamento das perspectivas da inteligência artificial são resultados de um esforço inicial proposto por Alan Turing em 1950 com o *The Imitation Game* [18], que tinha como um dos seus objetivos definir um conceito comum de IA não existente até então. Turing afirmou que:

> *"Não sabemos definir precisamente o que é inteligência e, consequentemente, não podemos definir o que é inteligência artificial. Entretanto, embora não tenhamos uma definição de inteligência, podemos assumir que o ser humano é inteligente. Portanto, se uma máquina fosse capaz de se comportar de tal forma que não pudéssemos distingui-la de um ser humano, essa máquina estaria demonstrado algum tipo de inteligência que, nesse caso, só poderia ser inteligência artificial."*

Em *The Imitation Game,* Turing elaborou um teste que hoje é considerado determinístico para aferir se uma máquina pode ser considerada ou não uma IA. Neste teste, Turing promoveu por meio de um terminal de diálogo, uma comunicação entre homem e máquina. Se após algumas perguntas e respostas o ser humano não conseguisse distinguir que estava dialogando com uma máquina, então, esta máquina poderia ser considerada verdadeiramente uma inteligência artificial. Ainda no artigo *The Imitation Game*, Turing trabalhou áreas de conhecimentos que hoje são consideradas a base da inteligência artificial, são elas:

- **Processamento de linguagem natural (PLN):** Representa a tentativa de extrair uma representação de significado mais completo de um texto. O PLN tipicamente faz uso de conceitos linguísticos, tal como a exploração da estrutura gramatical de um texto [9]. Esta área de estudo é um dos focos deste trabalho e será melhor detalhada na seção 2.3.

- **Representação de conhecimento:** Estuda como o conhecimento pode ser representado e manipulado simbolicamente de forma automatizada por programas de raciocínio. Mais informalmente, é a parte da AI que se preocupa com o pensamento, e como esses podem contribuir para o comportamento inteligente [1].

- **Raciocínio automatizado:** Se refere a uma subárea da IA que estuda formas de simular raciocínio lógico por meio de métodos computacionais [13].

- **Aprendizado de máquina:** Aprendizagem de máquina está na interseção de ciência da computação, engenharia e estatística. Uma ferramenta que pode ser aplicada a muitos problemas. Qualquer campo que precise interpretar e agir sobre dados pode se beneficiar de técnicas de aprendizagem de máquina. [7].

2.2 Chatterbots

Fundamentados pelo Teste de Turing, os *chatterbots*, são ferramentas *software* capazes de propor um diálogo com o ser humano em linguagem natural. Este diálogo se dá por meio da capacidade de compreensão e simulação do seu comportamento. Simon Laven definiu os chatterbots como [9] , *"... um programa que tenta simular a conversa escrita, com o interlocutor, tentando, pelo menos temporariamente, se fazer passar por um ser humano"*. Esta percepção vem se sustentando até os dias de hoje, o objetivo dos *chatterbots* se misturam com os objetivos da inteligência artificial, no que se refere a proximidade do comportamento humano, pois esta é até hoje uma referência incontestável de inteligência.

2.2.1 Contexto histórico

Com o passar dos anos e com a evolução da tecnologia, os *chatterbots* se modelaram aos novos conceitos vigentes. Podemos analisar a evolução dessas ferramentas por uma visão histórica das aplicações que foram desenvolvidas e que marcam a sua geração, como [2,3]:

- **Primeira Geração:** Os *chatterbots* da primeira geração são caracterizados pela utilização de regras gramaticais simples e casamento de padrões. Uma técnica que visa identificar se em um frase escrita pelo interlocutor possui uma estrutura específica, com o objetivo de conceber resposta aos padrões encontrados. O *chatterbot* ELIZA criado em 1966 por Joseph Weizenbaum, é um dos precursores desta geração e seu papel era simular um psicanalista rogeriano[2] em uma conversa com um paciente, o instigando com perguntas sobre as questões que ele mesmo levanta.

- **Segunda Geração:** A segunda geração, baseada em técnicas de inteligência artificial, como redes neurais é lembrada pelo JULIA, de Michael Mauldin. Este *chatterbot* representava um personagem dentro de um jogo de computador de multi-interlocutors. Seu objetivo era de simular um jogador que auxiliava os demais jogadores durante o decorrer do desfecho do jogo.

- **Terceira Geração:** A terceira e mais recente geração de *chatterbots* é baseada no uso de linguagens de marcação para a construção de bases de conhecimento.

[2] "A Abordagem Centrada na Pessoa (ACP)" https://psicologado.com/abordagens/centrada-na-pessoa/a-abordagem-centrada-na-pessoa-acp . Acessado em 11 dez. 2016.

Utilizam a linguagem AIML (Artificial Intelligence Markup Language). Seu maior representante é A.L.I.C.E (Artificial Linguistic Internet Computer Entity).

Este trabalho difere em parte das gerações e propõe uma nova perspectiva de construção de *chatterbots* como um sistema de recuperação da informação, que se mostra como uma novo segmento para construção de bots [16]. O Objetivo é de propor um método adaptativo para construção de *bots* baseado em práticas de PLN.

2.2.2 Características e Classificações

Analisando como os *chatterbots* surgiram, se faz é necessário uma melhor compreensão sobre quais características podem ser consideradas fundamentais em um olhar generalista sobre tudo aquilo que um *bot* pode representar. Como podemos ver [11]:

- **Autonomia:** Esta é uma das principal característica para um *bot,* se refere a capacidade que ele tem sobre o controle de suas próprias ações, de modo que a interferência humana seja mínima ou inexistente.

- **Pró-atividade:** É esperado que o *bot* responda aos estímulos da conversação, por tanto é de sua responsabilidade prover a iniciativa necessária para estimular a conversar e atingir os seus objetivos.

- **Continuidade temporal:** Diz respeito a contínua disponibilidade do *bot* sobre os seus serviços.

- **Capacidade Social:** É a capacidade que o *bot* tem de se comunicar com os seres humanos ou com até mesmo outros *bots*.

- **Capacidade de adaptação:** Fala sobre a capacidade de adaptação de um *bot* em relação a mudança de seu comportamento com base na experiência. A adaptação pode ser relativa ao ambiente ou no sentido de melhorar a sua interação com o interlocutor.

- **Flexibilidade:** Um *bot* com flexibilidade não executa ações pré-definidas, pois possui a capacidade de escolher dinamicamente as ações e a sequência de ações em resposta a um estado do ambiente.

- **Caráter:** Possui personalidade e estado emocional.

Em uma visão da representatividade de um *bot*, é possível analisar em sua maior parte características e essências humanas, também fazendo parte o atendimento à domínios de conhecimentos que podem variar de acordo com sua finalidade. Existem hoje diversos tipos de bots sobre os mais variados domínios de conhecimento, como pode ser visto na Tabela 2.

Categoria	Características
Academic bots	Relacionados aos assuntos acadêmicos como sites de professores ou laboratórios
Design bots	Possuem ferramentas e habilidades para a produção de outros bots e agentes inteligentes
Commerce bots	Desempenham atividades de comércio na internet
Fun bots	Divertem interlocutors através de jogos, ambientes virtuais e personagens de realidade virtual
Government bots	Buscam informações em site governamentais
Knowledge bots	Congrega agentes inteligentes, agentes de informação, agentes da *web* e muitas ferramentas inteligentes de busca
News bots	Criam jornais personalizados e clips de artigos de jornais do mundo inteiro
Search bots	Bots e agentes inteligentes de busca na Internet
Shopping bots	Fazem compras e comparações de preços para internautas
Stock bots	Monitoram o mercado de ações e mandam mensagens sobre os últimos preços e tendências
Update bots	Avisam ao interlocutor quando um site especificado foi atualizado ou modificado
Chatter bots	São programas que simulam uma conversa com um ser humano

***Tabela 2** - Categoria de bots [10]*

Naturalmente os *bots* podem ter um ou mais domínios e sua base de conhecimento [3], é também uma característica de um *bot* o auto conhecimento, sendo capaz de responder perguntas como: "Qual o seu nome?" e "Qual o seu propósito?".

2.2.3 Modelos de aquisição de conhecimento

Em termos de base de conhecimento, um dos desafios para os *bots* está no processo de aquisição de novos conhecimentos, um *bot* está hoje limitado ao seu contexto de domínio, isso diz respeito à capacidade que ele tem em conseguir suprir a necessidade de conhecimento do interlocutor sobre o domínio ao qual o *bot* foi desenvolvido. Em muitas vezes o *bot* acaba por não atender as expectativas por não possuir uma base de

[3] "Base de conhecimento – Wikipédia, a enciclopédia livre." https://pt.wikipedia.org/wiki/Base_de_conhecimento. Acessado em 11 dez. 2016.

conhecimento ampla, sendo assim, as técnicas de aquisição de conhecimento servem para este propósito. São elas [14]:

- **Aquisição natural de conhecimento:** Essa forma provê que um *bot* possa adquirir conhecimento de maneira natural, ou seja, por meio do diálogo que é realizado. Contudo, isso requer um mecanismo que identifique que o interlocutor está propondo um novo conhecimento.

- **Aquisição impositiva de conhecimento:** Uma outra maneira é representada pela inserção direta na base de conhecimento. Nesta técnica o conhecimento contido na base de dados da aplicação se dá de maneira impositiva pelo desenvolvedor ou detentor do *bot*.

- **Extração do conhecimento da *Web***: É comum a utilização de mecanismo extratores de conteúdo da *web*, como fóruns e *Wikis*. Neste tipo de aquisição é utilizado algum sistema de extração de conteúdo ou até mesmo adotada a técnica de *web scraping*[4], que visa extrair de uma página *web* conteúdos de relevância que posteriormente serão moldados para que se adequem à estrutura da base de dados.

2.3 Processamento de Linguagem Natural

Podemos aferir que as áreas de conhecimentos levantadas por Turing e sobre o seu teste envolvendo *chatterbots* encadearam uma enorme quantidade de pesquisas na IA, em função do objetivo deste trabalho, PLN em uma forma mais detalhada, trata-se da área de conhecimento que em conjunto com a linguística, estuda os problemas da geração e compreensão automática de línguas humanas naturais, observando aspectos de comunicação humana como sons, palavras, sentenças e discursos, considerando formatos e referências, estruturas e significados, contextos e usos [1]. Diante disso, este trabalho irá realizar um levantamento sobre seus conceitos.

2.3.1 Níveis de entendimento

O PLN não é algo simples, devido a difícil tarefa de fazer a máquina compreender e se comunicar sobre a complexidade e ambiguidade de linguagem natural. Contudo, a fim de solucionar esta problemática o PLN fixa as suas raízes para estudar os mais diferentes níveis de entendimento da linguagem, que são [5, 22]:

- **Fonético:** Parte da linguística focada no estudado e nas variedades de relacionamento das palavras e os sons que elas produzem no ato da fala.

[4] "Web scraping - Wikipedia." https://en.wikipedia.org/wiki/Web_scraping. Acessado em 11 dez. 2016.

- **Morfológico:** Diz respeito a identificação das palavras em uma sentença. Analisando as suas unidades de significado primitivo e na classificação em categorias morfológicas.

- **Sintático:** Semelhante ao nível morfológico, o nível sintático define a estrutura de uma frase, sobre o papel individual de cada palavra em um sentença.

- **Semântico:** Estuda o relacionamento das palavras com seus significados e de como eles são combinados para formar os significados das sentenças.

- **Pragmático:** Parte da linguística que estuda a linguagem no contexto de sua utilização, o uso de frases e sentenças em diferentes contextos, afetando o significado.

2.3.2 Processamento

2.3.2.1 Aplicações

Sendo o respaldo linguístico de fundamental importância para o PLN, é também necessário compreender a utilização técnica desses conhecimentos em aplicações reais. Esta seção do trabalho será abordada uma síntese sobre os tipos mais comuns de aplicações de PLN. São eles [6]:

- **Sistemas de sumarização automática:** Esses sistemas caracterizam-se por gerar “extratos” (justaposição de porções do texto fonte) ou “resumos” (texto gerado a partir de um plano de resumo) de um ou mais textos de acordo, por exemplo, com uma determinada taxa de compressão.

- **Sistemas de tradução automática:** Partem de um texto fonte, escrito em uma língua natural x, e produzem um texto alvo, ou seja, uma versão do texto fonte em uma língua y. A adoção de sistemas de tradução automática em *bots* aumenta a abrangência de atendimento novas línguas naturais.

Dentre as aplicações de PLN, este trabalho irá implementar um **Sistemas de recuperação de informação (RI),** estes sistemas tratam essencialmente de indexação, busca e classificação de documentos textuais, com o objetivo de satisfazer a necessidade da informação expressas através de consultas [1]. Dentre os componentes essenciais de um RI estão o interlocutor, a necessidade da informação e uma coleção de documentos. Estes documentos podem ser representados por qualquer fonte de acesso à textos, seja um página *Web*, um arquivo de texto ou a representação de ambos em uma base de dados.

A representação do processo de recuperação da informação pode ser vista na Figura 1, onde é possível observar que a interface com o interlocutor e o sistema de

gerenciamento de base de dados (SGBD) para a coleção de documentos compõem as extremidades do processo. Contudo, é necessário atenuar para todo o fluxo que ocorre entre esses dois componentes, onde a necessidade do interlocutor é convertida em termo lógico entendível pela máquina, resultando em busca pelos documentos que condizem com esse termo e que já se encontram indexados de maneira ordenada devido a utilização da técnica de arquivo ou lista invertida[5].

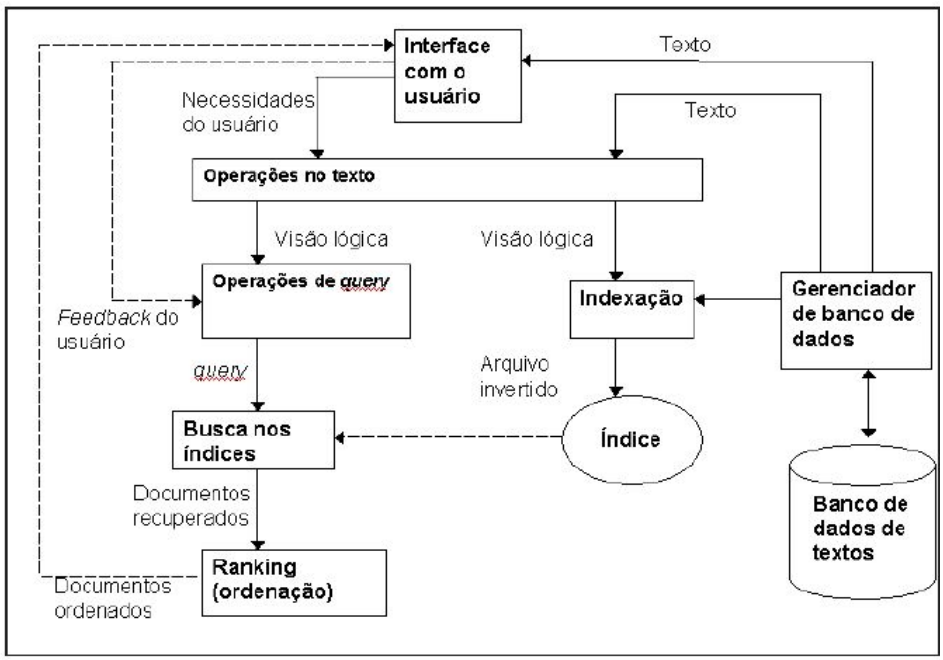

***Figura 1** - O processo de recuperação da informação (Adaptado de Baeza Yates & Ribeiro Neto, 1999, p, 10)*

A aplicação real de um modelo de processo de RI proposto sobre a realidade dos bots pode ser visto mais detalhadamente na seção 3.3.

2.3.2.2 Estratégias

As aplicações em PLN mesmo tendo objetivos distintos, acabam por utilizar estratégias semelhantes para obter um melhor desempenho, essas estratégias são adotadas no decorrer do processamento de linguagem natural. Aqui serão abordadas as estratégias mais comuns e que podem ser aplicadas para fins de construção de um RI. São eles [6]:

- **Etiquetagem:** *part-of-speech tagger* ou etiquetador gramatical, que diz respeito à técnica aplicada de se atribuir etiquetas a um texto, a fim de facilitar a aplicação do conhecimento linguístico. Existem etiquetadores morfológicos, aquelas aos quais definem categorias morfológicas como substantivos e adjetivos, como etiquetadores sintáticos, aquelas aos quais atribuem indicações sintáticas como sujeito e objeto, e etiquetadores semânticos, que indicam os papéis dos itens lexicais na sentença, como agente, processo e estado.

[5] "Listas invertidas – Wikipédia, a enciclopédia livre." https://pt.wikipedia.org/wiki/Listas_invertidas. Acessado em 8 dez. 2016.

- **Normalização morfológica (Stemming):** Ocorre na redução dos itens lexicais através de conflação, que significa a combinação da representação de dois ou mais termos em um único termo, utilizando a remoção dos sufixos e prefixos das palavras resultando em sua forma radical. A utilizando da normalização é conhecida como pelo morfológica em um *bot* impacta na diminuição da variação das palavras, o que leva ao melhor processamento comparativo. Em caso da utilização da técnica de casamento de padrão, esse procedimento pode fazer com que mais sentenças possam se adequar ao mesmo padrão escrito.

- **Eliminação dos *stopwords*:** Os *stopwords* são palavras funcionais, como artigos, conectivos e preposições. Estes itens léxicos muitas vezes acabam tirando o poder de performance do processo de linguagem natural. A adoção dessa técnica em *bots* da terceira geração por exemplo, pode resultar em uma melhora expressiva no casamento das regras (gatilhos) contidos em arquivos .xml.

- **Tokenização:** Esta técnica consiste na quebra de um texto em segmentos chamados *tokens*, estes segmentos são unidades linguísticas, tais como palavras, sinais de pontuação ou números alfanuméricos. Em PLN este processo é necessário para realizar uma análise morfológica sobre uma palavra (segmento). O exemplo abaixo mostra como ocorre o processo de tokenização:

  Frase: "A casa 16 é amarela e bonita!"

A	casa	16	é	amarela	e	bonita	!

- **Aplicação de métodos estatísticos:** Os métodos estatísticos são muitos, eles surgem da necessidade da aferição de problemáticas específicas, como por exemplo, a determinação por meio matemático de um coeficiente de importância de uma palavra dentro de um texto, medido pela quantidade de repetições que a palavra obteve em uma frase. Um dos grandes desafios para os *bots* está em mensurar o quanto uma palavra pode ser importante dentro de frase, com esse conhecimento ele pode aferir qual possível resposta pode atingir melhor as expectativas de um interlocutor.

Diante da vasta quantidade de técnicas e estratégias a serem aplicadas em sistemas de PLN é necessário uma avaliação mais detalhadas de quais combinações de técnicas podem resultar em no alcance mais concreto do objetivo traçado pelo sistema de PLN.

2.4 Considerações finais

Observando conceitos sobre *chatterbots,* inteligência artificial e a gama de técnicas e estratégias atreladas a PLN, assim como, seus níveis de entendimento, aplicações e tipos de processamento dizem respeito a conhecimentos fundamentais para a construção da aplicação ao qual este trabalho se propõe. É possível também compreender que o uso de Sistemas de Recuperação da Informação dizem respeito ao conjunto destas técnicas com o objetivo real e comum da necessidade da informação.

O próximo capítulo visa apresentar a aplicação proposta em comparativo com os assunto abordados até o presente momento, realizando análise e apresentação de modelos adaptativos à realidade dos bots.

Capítulo 3

3. Usain Bots

Este capítulo abordará a proposta proposta de serviço do Usain Bots, demonstrando suas características, arquitetura e conceitos que auxiliam a construção de *bots*.

3.1 Proposta

O Usain Bots trata-se de um serviço *web* sobre o conceito *Open Source* destinado ao gerenciamento e processamento lógico de *bots* que são projetados para atender as necessidades e características essenciais dos mesmos. São algumas características do serviço:

- Arquitetura de dados para gerenciamento de *bots* e aplicações cliente;
- Aplicação de PLN em um Sistema de Recuperação da Informação;
- Mecanismo de controle de acesso;

3.1.1 Aplicação das características de um *bot*

Dentre as características que dizem respeito a essência de um *bot* e que podem ser vistas na seção 2.2.1, a aplicação proposta compreende:

- **Autonomia**: A autonomia do serviço está na classificação de maneira autónoma de intenções baseada um treinamento supervisionado.
- **Pró-atividade**: A aplicação tem como um de seus objetivos o buscar estímulo da conversa por meio de sugestões de novos assuntos e através de questionamento realizados ao interlocutor.
- **Continuidade temporal:** Sendo esta uma importante característica de um *bot*, o serviço proposto conta com um tratamento de falhas que possui o objetivo de minimizar possíveis interrupções no serviço.
- **Capacidade Social:** Por meio de um protocolo de comunicação web HTTP a aplicação provê entre seus serviços a troca de interação social com interlocutors de aplicações clientes. Atendendo a estímulos de interação que estão fora da base de conhecimento específica.

3.1.2 Modelo de aquisição de conhecimento

Dentre os modelos de aquisição de conhecimento referenciados na seção 2.2.3, foi adotado o modelo de aquisição impositiva do conhecimento e a extração do conhecimento da *Web*. A aplicação destes modelos compreendem a base de conhecimento de um *bot* no serviço. A carga de conhecimento se dá por meio de uma requisição HTTP para o serviço do Usain Bots seguindo a especificação contida na documentação[6].

Os arquivos de conhecimento são inseridos na base de dados seguindo um conceito de documento que pode ser visto na seção 3.3.2.3, sendo a responsabilidade do administrador do bot a sua atualização de base de conhecimento. Em especial, a extração do conhecimento da *Web* se mostra uma forte aliada pela busca da geração do conhecimento de maneira mais autônoma, minimizando a necessidade de uma intervenção tão direta e manual do administrador na produção de conhecimento.

3.2 Arquitetura

3.2.1 Node.js e Express

Atendendo necessidade da aplicação deste trabalho, a tecnologia Node.js foi adotada como ferramenta que irá suprir a necessidade de desenvolvimento de um serviço web. Criada por Ryan Dahl em 2009, o Node.js é uma plataforma *Open Source* escrita em JavaScript e que utiliza a engine V8 do navegador Google Chrome para desenvolvimento de aplicações *server-side* com o objetivo de auxiliar os desenvolvedores a criarem aplicações de rede de alta flexibilidade e escalabilidade. Contudo, a fim de diminuir o tempo de desenvolvimento de uma aplicação Node.js, este trabalho adotou também o *framework* Express. Este *framework* acrescenta ao Node.js alguns componentes essenciais para desenvolvimento de aplicações em rede. Como por exemplo, rastreamento de rotas, sistema de middlewares, gerenciamento de sessão e arquivos.

3.2.1.1 Arquitetura do Node.js

Para compreendermos o Node.Js se faz necessário o entendimento de sua arquitetura e características que o fazem. Como é possível ver na Figura 2. São eles:

- **E/S não bloqueante:** O Node.js trabalha sobre o conceito de solicitações de entrada e saída não bloqueantes. Desta forma, não existe a interrupção natural de todos os demais processos quando uma solicitação é recebida [17].

- **Single Thread:** O Node.js usa uma instância da aplicação para gerenciar todas as solicitações e conexões na rede. Quando o Node.js é inicializado, ele informa ao

[6] Documentação Usain Bots - http://docs.usainbot.apiary.io/

sistema operacional que ele deve ser notificado quando uma nova conexão é feita e assim ativa o modo de suspensão para economia de uso da CPU [17].

- **Event Loop:** É o agente responsável por escutar e emitir eventos da aplicação. Em poucas palavras ele é um loop infinito que a cada iteração verifica em sua fila de eventos se um determinado evento foi emitido [12].

- **Libuv:** Corresponde a uma biblioteca escrita em C++ responsável pelas operações de entrada/saída e pelo event loop. Ela é composta pela thread *pool* de tamanhos fixos, onde uma thread é alocada para cada operação de entrada/saída [17].

- **Callbacks:** São eventos de retorno, funções JavaScript que são colocadas em fila para serem chamadas quando permitidas, executando o fluxo da solicitação [12].

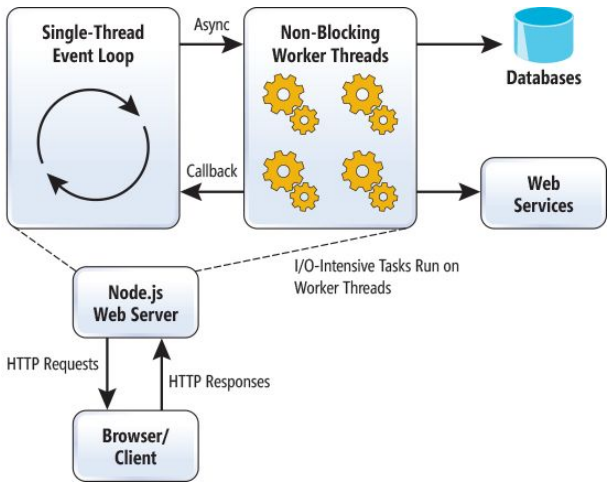

Figura 2 *- Orquestração de solicitação-resposta do Node.js* [17]

A figura acima demonstra o processo de solicitação e resposta no Node.js. Quando uma nova solicitação é registrada pelo Node.js, o *event loop* delega a solicitação para as *threads* de trabalho e registra os seus respectivos *callbacks*. A partir disso, processos de controle da aplicação, conexão à base de dados e solicitação a novos serviços *web* são liberados e executados de maneira assíncrona, permitindo que o *event loop* está sempre disponível para receber novas solicitações.

3.2.1.2 Arquitetura MVC com Node.js e Express

Apesar de não possuir um modelo arquitetural rígido, o Node.js em conjunto com o Express permitem a utilização do modelo de arquitetura de software MVC (*Model*, *View*, *Controller*). Uma característica forte que representa este modelo é a separação da camada lógica da aplicação da camada de apresentação. Onde cada elemento possui papéis e responsabilidades bem definidas e distintas. São alguns elementos arquitetura:

- **Model:** Responsável por definir e manipular os dados em Node.js, esta camada está diretamente ligada ao banco de dados escolhido.

- **View:** Representa a camada de apresentação do conteúdo, sendo a camada de interação com o cliente (interlocutor).

- **Controller:** Diz respeito a parte lógica da aplicação, tem como principal função executar o fluxo da aplicação e servir como intermediador entre o *model* e a *view*.

- **Dispatcher:** Este componente recebe as requisições HTTP e tem como função o processar essas solicitações de entrada e determina qual a rota correspondente a requisição será executada.

- **Router:** Diz respeito aos *end-points* (URIs). Este componente tem como responsabilidade acionar os middlewares atrelados definidos para cada rota do controlador.

Este trabalho irá compreender uma adaptação da arquitetura MVC para MC, uma vez que serviços *web* não exigem obrigatoriamente a implementação da camada de apresentação. Sendo assim, o modelo representativo do processo de solicitação e resposta é representado pela Figura 3.

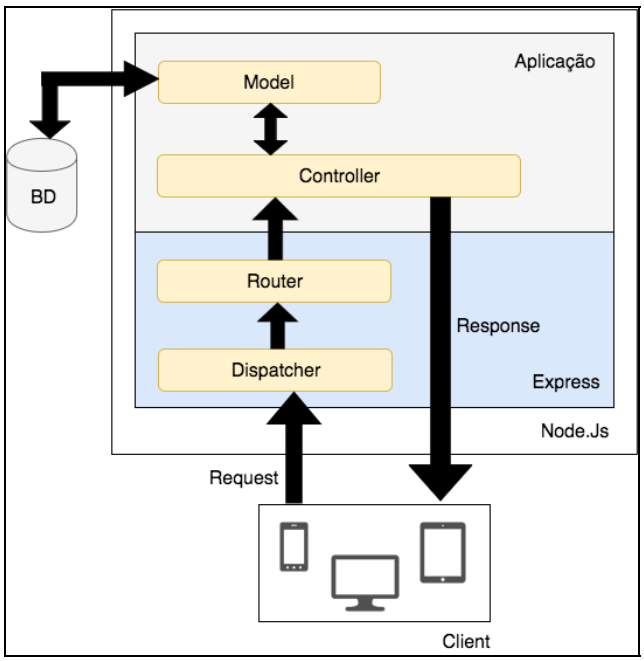

Figura 3 *- Modelo de solicitação e resposta do Usain Bots.*

3.3 Usain Bots

Sobre os conceitos de desenvolvimento de aplicações Node.js, esta seção demonstra a organização arquitetural do serviço Usain Bots, compreendendo seus principais módulos e componentes. A Figura 4 ilustra a arquitetura sobre um representação lógica.

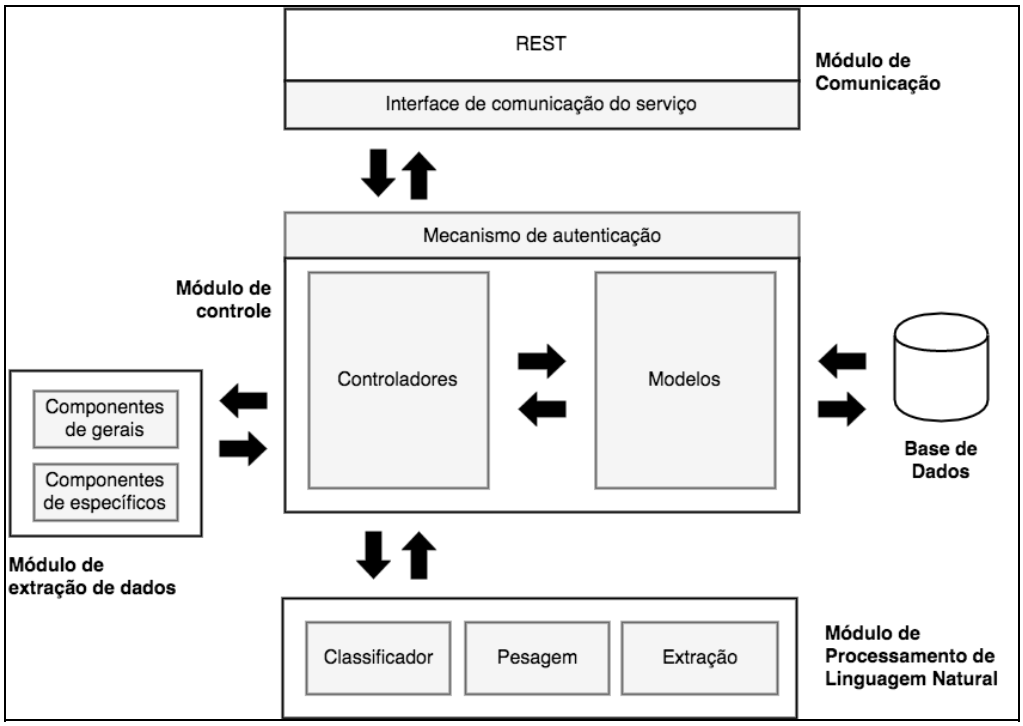

***Figura 4** - Representação lógica da arquitetura do Usain Bots*

Seguindo a continuidade desta leitura, as próximas seções irão apresentar cada módulo da arquitetura, ressaltando os seus papéis e características.

3.3.1 Módulo de Comunicação

O módulo de comunicação e acesso representa a entrada e saída para serviços externos, este módulo aplica o modelo REST[7] de *Application Programming Interface (API)* a fim de propor uma interface de comunicação padronizada para uso dos serviços do Usain Bots.

A comunicação ocorre por meio de requisições dos mais diferentes recursos, estas requisições são enviadas ao módulo de controle de acesso para que o fluxo lógico do serviço possa prosseguir ou não.

3.3.2 Módulo de Controle

Do ponto de vista lógico, este módulo representa o fluxo de controle do serviço, compreendido pelos *controladores* e *modelos* da representação da arquitetura MVC. Desta forma, os componentes deste módulo influem diretamente no acesso aos outros

[7] "REST – Wikipédia, a enciclopédia livre." https://pt.wikipedia.org/wiki/REST. Acessado em 12 dez. 2016.

módulos ou na recusa de acesso por parte de uma solicitação do cliente, podendo também existir uma comunicação de acesso entre si.

3.3.2.1 Mecanismo de autenticação

Em função da necessidade de segurança dos dados e respectivamente dos *bots*, o mecanismo de autenticação da API representa a aplicação do conceito de *JSON Web Token (JWT)* [8], que utiliza *tokens* como mecanismo de validação e autorização, os *tokens* representam um *hash* em *base64* seguindo o modelo *header*, *payload* e *signature*, onde todos estão notados em JSON. O componente *header* informa qual tipo de token está sendo gerado e qual algoritmo que foi utilizado para a criptografia, o *payload* corresponde ao dados que serão criptografados, e por fim, o *signature* é como o próprio nome diz, a assinatura do *hash* que não pode ser descriptografado e que armazena também a validade de expiração um *token*, sendo ele responsável então por garantir a integridade dos dados armazenados. Desta forma, a aplicação do *middleware* em rotas garante a segurança de acesso e gerenciamento do *bot*. O método isAuthenticated, visto na Figura 5, representa a função que faz o tratamento da autorização de acesso.

```
8  exports.isAuthenticated = function(req, res, next) {
9
10   var headers  = req.headers.authorization;
11   var token    = null;
12
13   if(!headers){
14     return res.status(400).json({
15       status: 400,
16       erro: 'bad_request',
17       mensagem: 'Erro de requisição'
18     });
19   }
20
21   var bearer = headers.split(' ')[0].toLowerCase().trim();
22
23   if(bearer != "bearer") {
24     return res.status(400).json({
25       status: 400,
26       erro: 'bad_request',
27       mensagem: 'Erro de requisição'
28     });
29   }
```

Figura 5 *- Arquivo auth.js e método isAuthenticated*

Em termos de aplicação cliente, foi aplicado um método extra de segurança, onde cada aplicação cliente possui um *secret*, que também corresponde a um *hash* em *base64*. Este é uma *hash* gerado de maneira independente sem validade de expiração e representa a identificação mais segura de uma aplicação cliente.

3.3.2.2 Controladores

Os controladores são componentes de fluxo lógico do serviço, capaz de orquestrar e acionar os demais módulos, estes são representados também por um mecanismos de rastreamento de rotas como pode ser visto na Figura 6.

```
19  router.post('/bot/cadastrar',                auth.isAuthenticated, cadastrar);
20  router.put('/bot/:id/atualizar',             auth.isAuthenticated, atualizar);
21  router.get('/bot/:id/consulta',              auth.isAuthenticated, consultar);
22  router.delete('/bot/:id/remover',            auth.isAuthenticated, remover);
23  router.post('/bot/:id/documento/cadastrar',  auth.isAuthenticated, cadastrarDocumento);
24
25  function cadastrar (req, res, next) {
26
27    if(req.body.dados) {
28
29      var dadosReq = req.body.dados;
```

Figura 6 *- Mapeamento de rotas do controlador bot.js*

3.3.2.3 Modelos e Base de dados

Os modelos em uma arquitetura MVC estão ligados à base de dados. Desta forma, uma boa aderência entre o modelo de dados adotado e a tecnologia do serviço é de fundamental importância. Sendo assim, a escolha pelo SGBD se fundamentou na análise dos critérios de dinamismo, flexibilidade e melhor compatibilidade com JavaScript (Node.js). Seguindo estes critérios foi adotado um SGBD não SQL representado neste trabalho pelo MongoDB[8], um dos mais relevantes SGBDs não SQL, que está no entre os 5 SGBDs mais populares no mundo, sendo primeiro lugar entre os SGBDs não SQL [4].

A compreensão dos modelos no MongoDB é representada através de esquemas que tem como função determinar os atributos de uma coleção de dados juntamente com seus tipos e características. São modelos aplicados neste trabalho:

- **Modelo Aplicação**

  O Modelo Aplicação apresentado na Figura 7, é um coleção de dados que representa uma aplicação cliente seguindo critérios de segurança vistos na seção 3.3.2.1. Este modelo é composto pelo *id* da aplicação gerado de forma automática pelo MongoDB, nome e pelas credenciais de acesso *secret* e *token*.

```
5  var aplicacaoSchema = new mongoose.Schema({
6    nome:           { type: String, require: true },
7    secret:         { type: String, require: true, unique: true },
8    token:          { type: String }
9  });
```

[8] "MongoDB for GIANT Ideas | MongoDB." https://www.mongodb.com/. Acessado em 9 dez. 2016.

***Figura 7** - Esquema MongoDB da coleção de dados "Aplicação"*

- **Modelo Bot**

O Modelo Bot apresentado na Figura 8, é um coleção de dados que representa um *bot*, compreendido pelos atributos *idApp,* que faz referência ao id da coleção Aplicação, o nome do *bot*, a data da criação da coleção e frases de abertura, agradecimento, encerramento, engajamento e sem resposta, que auxiliam no processo de PLN.

```
var botSchema = new mongoose.Schema({
  idApp:              { type: String, require: true },
  nome:               { type: String, require: true },
  dataCriacao:        { type: Date },
  frases: {
    abertura:               [ String ],
    agradecimento:          [ String ],
    encerramento:           [ String ],
    semResposta:            [ String ],
    engajamento:            [ String ]
  }
});
```

***Figura 8** - Esquema MongoDB da coleção de dados "Bot"*

- **Modelo Documento**

O modelo Documento, visto na Figura 10 representa um conceito de documento compreendido pela Figura 9. Para este trabalho um documento é uma parte de um todo, considerando o todo como qualquer tipo de texto contendo um título e conteúdos, neste caso, podemos enxergar um documento como uma fração composta pelo título e o conteúdo de um texto com *n* títulos e *n* conteúdos.

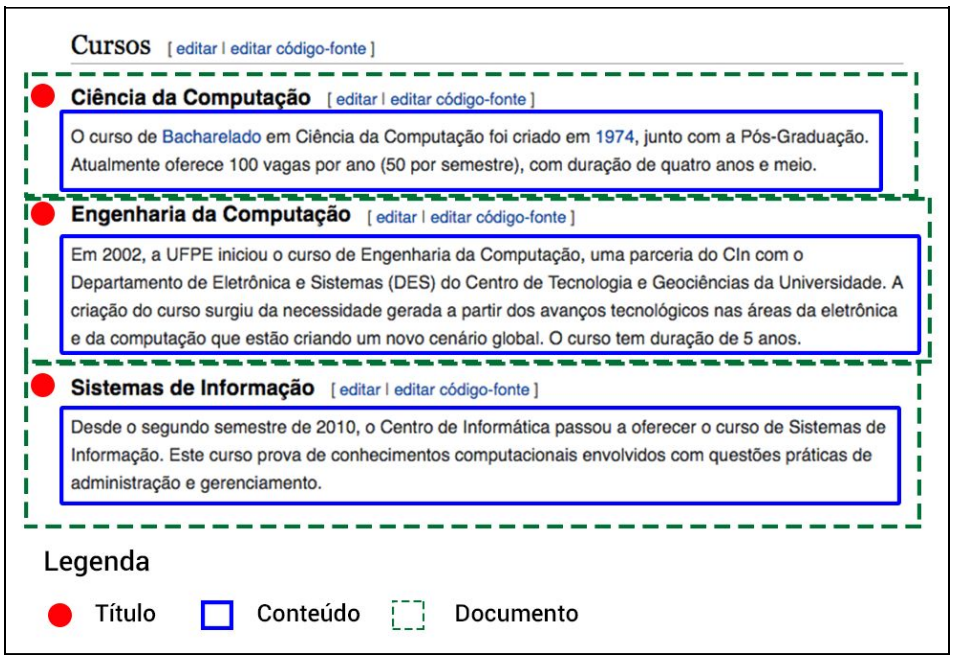

Figura 9 *- Representação visual de um "Documento"*

O esquema de um documento, representado pela Figura 12, é composto pelo seu *id*, *idBot* que é considerado a chave de identificação da coleção de dados Bot, o atributo *tags que* diz respeito ao modelo de indexação de um documento, o atributo *dataCriacao* representa a o momento em que o documento foi gerado, e os atributos *uri* e *seletor* são derivados a partir de uma coleta a uma página *web*, se referindo respectivamente ao endereço da página e ao nível de profundidade de uma elemento HTML em uma árvore DOM[9].

```
5  var documentoSchema = new mongoose.Schema({
6    idBot:        { type: String, require: true },
7    tipo:         { type: String, require: true },
8    uri:          { type: String },
9    seletor:      { type: String },
10   titulo:       { type: String },
11   conteudo:     [ String ],
12   tags: {
13     titulo:   [ String ],
14     conteudo: []
15   },
16   dataCriacao:  { type: Date, require: true }
17 });
```

Figura 10 *- Esquema MongoDB da coleção de dados "Documento"*

- **Modelo Evento e Notícia**

Estes modelos são coleções de dados específicas para o CIn Bot e representam a coleção de notícias e eventos extraídos do site institucional do departamento. As Figuras 11 e 12 apresentam o modelo de dados definidos para estas coleções e que são compostas pelo chave de identificação do modelo de dados Bot, título da notícia ou do evento, o endereço *uri* da página de extração do conteúdo, data da postagem do evento e resumo da notícia.

```
4  var eventoSchema = new mongoose.Schema({
5    idBot:          { type: String, require: true },
6    titulo:         { type: String, require: true },
7    data:           { type: Date,   require: true },
8    uri:            { type: String, require: true }
9  });
```

Figura 11 *- Esquema MongoDB da coleção de dados "Evento"*

[9] "Modelo de Objeto de Documentos – Wikipédia, a enciclopédia livre." https://pt.wikipedia.org/wiki/Modelo_de_Objeto_de_Documentos. Acessado em 9 dez. 2016.

```
var noticiaSchema = new mongoose.Schema({
  idBot:          { type: String, require: true },
  titulo:         { type: String, require: true },
  resumo:         { type: String, require: true },
  uri:            { type: String, require: true }
});
```

Figura 12 *- Esquema MongoDB da coleção de dados "Notícia"*

3.3.3 Módulo de Processamento de Linguagem Natural

O Módulo de PLN representa a junção dos componentes que formam o processamento textual e de recuperação da informação. Estes estão descritos pelos componentes:

3.3.3.1 Extração

O componente de extração tem como finalidade atuar sobre o processamento textual, aplicando métodos do PLN para extrair de uma frase apenas elementos que possam ser aproveitados para um processo de indexação de um documento ou por um tratamento de de classificação de uma frase. As etapas deste processo podem ser vista na Figura 13 e melhor representadas conceitualmente na seção 2.3.

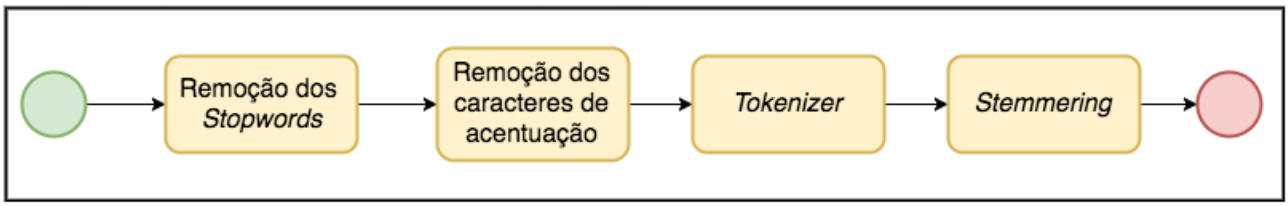

Figura 13 *- Processamento de Linguagem Natural*

3.3.3.2 Classificador

O componente Classificador compreende a necessidade de identificação da classificação de uma frase provinda de um interlocutor a fim de determinar qual a intenção do mesmo e prover a continuidade do fluxo lógico para o tratamento de cada uma das seguintes categorias:

- **Abertura**: Identifica que a frase é possivelmente uma saudação
- **Agradecimento**: Identifica que a frase é se trata de um agradecimento
- **Questionamento**: Identifica que a frase é possivelmente um questionamento
- **Encerramento**: Identifica que a frase se trata de um encerramento de diálogo
- **Cumprimento**: Identifica que a frase se trata de um cumprimento por parte do cliente.

Este procedimento ocorre pela aplicação do algoritmo classificador Naive Bayes[10], que por meio de um treinamento supervisionado e pela aplicação de métodos estatísticos encontrados no Teorema de Bayes[11] consegue determinar padrões nos texto que possibilitam determinar a categoria de novos textos. Em poucas palavras, o processo ocorre na inserção de *n* entradas que correspondem a frases e suas respectivas categorias previamente definidas, após o treinamento, o algoritmo está apto a determinar sobre qual categoria uma *frase* pertence. Este algoritmo foi implementado no Usain Bots com o objetivo de reconhecimento de intenção, a fim de determinar qual a classificação de um *frase* de entrada e prover comportamentos distintos para cada classificação.

3.3.3.3 Pesagem

Este componente tem como função a análise de uma série de documentos provindos da base de dados e que serão pesados para determinar qual documento atende melhor as necessidade de um interlocutor. Sendo assim, um documento com título e conteúdo(s) passam previamente pelo componente de extração com o objetivo de gerar suas respectivas *tags*, essas *tags* correspondem ao indexador do próprio documento e são elas que irão auxiliar no processo da recuperação da informação. Sentindo a necessidade de buscar uma frase recém processada, a consulta ao banco de dados se dá da verificação igualitária entre frase e *tags*.

Uma vez coletado os documentos que correspondem uma *frase*, é aplicado sobre cada documento recuperado do banco de dados um mapeamento de peso pelo algoritmo Td-idf, uma medida estatística que avalia o quão importante uma palavra é um documento de acordo com a frequência com que ela se repete. Desta forma, ocorre uma verificação do peso de cada palavra de uma *frase* e cada um dos documentos recém recuperados. Posteriormente ainda é aplicado uma média ponderada para título e conteúdos de cada documento, onde título corresponde ao peso 2 e conteúdo ao peso 1. Isso se torna necessário para suavizar a disparidade de peso que um documento com maior conteúdo tem sobre um documento com menor conteúdo. A fórmula representa o cálculo final de um peso:

$$\mathbf{PD} = \left(\frac{(2 \times (T / QT)) + (\sum_{i=1}^{n} 1 \times (C_i / QC_i))}{3} \right)$$

Onde,

PD: Peso resultante de um documento *n*.

[10] "Naive Bayes classifier - Wikipedia." https://en.wikipedia.org/wiki/Naive_Bayes_classifier. Acessado em 9 dez. 2016.
[11] "Teorema de Bayes – Wikipédia, a enciclopédia livre." https://pt.wikipedia.org/wiki/Teorema_de_Bayes. Acessado em 9 dez. 2016.

T : Peso resultante do algoritmo Td-Idf para o título no mesmo documento *n*.

QT: Quantidade de palavras de um título do documento *n*.

C*i* : Peso resultante do algoritmo Td-Idf para conteúdo n do mesmo documento *i*.

QC*i* : Quantidade de palavras de um conteúdo *i*.

Após isso, o algoritmo retorna a frase que possui o maior peso, entendendo-se que esta possa atender melhor às necessidades do interlocutor.

3.3.4 Módulo de extração de dados

O módulo de extração de dados representa os componentes que atuam sobre a aquisição do conhecimento de um *bot* a partir da extração do conhecimento da *web*. Este módulo conta com coletores de uso geral, aqueles aos quais podem ser usados por qualquer *bot* que pretenda extrair conteúdo da web, como também os coletores específicos, criados apenas para a construção do CIn Bot. Em exemplo pode ser visto na Figura 14 e na Figura 15 uma das funções que correspondem aos coletores gerais e específicos respectivamente.

```
62  exports.getContentWiki = function(data, done){
63
64    if(data.charset != 'utf8') {
65      data.encoding = null;
66    }
67
68    request(data, function(err, res, body) {
69
70      if(data.charset != 'utf8') {
71        body = iconv.decode(body, data.charset);
72      }
73
74       var $ = cheerio.load(body);
75
76       var p = [];
77
78       var $extractor = $('#mw-content-text');
79       var $h2        = $extractor.find('h2');
80       var $h3        = $extractor.find('h3');
81       var titulo     = "";
82       var conteudo   = [];
83       var extract    = [];
84
85       $h2.each(function(i, elem) {
86
87         titulo = $(this).children('.mw-headline').text().trim();
88         conteudo = $(this).next('p').text().trim();
```

Figura 14 *- Método getContentWiki do coletor geral*

```
exports.getNews = function(done, config){

  request(config, function(err, res, body) {
    if(err) return;

    var bodyEncoded = iconv.decode(body, 'iso-8859-1');

    if(res.statusCode === 200) {

      var $ = cheerio.load(bodyEncoded);
      var noticias = [];

      $('ul[class=listaNoticia] a').each(function(i, elem) {

        if(i < 10) {
          noticias.push({
            titulo: $(this).children('h3').text().trim(),
            resumo: $(this).children('p').text().trim(),
            uri:    config.url + $(this).attr('href'),
          });
        }
```

Figura 15 *- Método getNews do coletor específico do CIn Bot*

3.4 Considerações finais

Este capítulo apresentou o conceito do Usain Bots em conjunto com sua arquitetura e implementação, que busca facilitar o desenvolvimento de *bots* por meio de um serviço *web*, respeitando a essência de um *bot* e provendo componentes que se adaptam às suas necessidades.

O próximo capítulo visa explicar o uso dos recursos propostos pelo Usain Bots em função da construção da aplicação cliente CIn Bot.

4. Utilização do Usain Bots na construção do bot do CIn

Este capítulo abordará a aplicação de uso do serviço web Usain Bots em uma interface cliente do Centro de Informática da Universidade Federal de Pernambuco, sendo assim, foi realizada a construção de uma aplicação *web* denominada CIn bot que irá consumir os dados provindos pelo serviço proposto por este trabalho.

4.1 Arquitetura

A aplicação CIn Bot foi desenvolvida também em uma arquitetura MVC e se deu pela adoção do framework *front-end* Angular.js[12], escolhido por sua praticidade de implementação e por abranger um cliente HTTP embutido que facilita a comunicação a serviços externos como o do Usain Bots.

4.1.2 Controlador

O CIn bot conta com dois controladores com funções e responsabilidades distintas, são eles:

- **App**: Diz respeito ao controlador geral da aplicação, este compreende as configurações básicas para funcionamento da aplicação, como injeção de módulos e rastreamento de rotas. Como pode ser visto na Figura 16.

```
6  angular.module('usainBot').config(function($routeProvider, $locationProvider) {
7
8      $routeProvider.when("/chat", {
9          templateUrl : "app/views/chat.html",
10         controller: "Chat"
11     }).otherwise({
12         redirectTo: '/chat'
13     });
14
15 });
16
```

Figura 16 - *Controlador app.js do CIn Bot*

- **Chat:** O controlador Chat é responsável por aplicar o fluxo lógico do CIn Bot, com solicitações ao serviço Usain Bots e injetando os objetos JavaScript na View *chat.html*. Este pode ser representado pela Figura 17.

[12] "AngularJS — Superheroic JavaScript MVW Framework." https://angularjs.org/. Acessado em 12 dez. 2016.

```
angular.module('cinBot').controller('Chat', chatController);

chatController.$inject = ['$scope', '$sce', '$timeout','CONFIG', '$http','$location', '$window'];

function chatController ($scope, $sce, $timeout, CONFIG, $http, $location, $window) {

  function consultar (mensagem, done) {

    $http({
      method: 'GET',
      url: CONFIG.api.host + '/bot/' + CONFIG.api.id + '/consulta/?q=' + mensagem,
      headers: {
        'Content-Type' : 'application/json'
      },
    }).then(function successCallback(res) {

      if(res.status = 200) {
        done(false, res.data);
      } else {
        done(true);
      }

    }, function errorCallback(res) {
      done(true);
    });

  }
```

Figura 17 - *Controlador Chat.js do CIn Bot*

4.1.3 View

No que se a camada de apresentação, o CIn bot contempla a *view* chat.html, representada na Figura 18 sobre uma estrutura *html* necessária para uma *interface* de um terminal de diálogo, compreendendo estruturas e notações de repetições.

```
<div class="scroll" id="scroll" scroll-glue-bottom="glued">
  <div ng-repeat="mensagem in mensagens" class="container-mensagem">

    <div ng-if="mensagem.tipo =='cinbot'" class="containerMensagem row animated fadeIn">
      <div class="col-xs-10 no-padding">
        <div class="mensagem cinbot">
          <p class='id'>CIn bot:</p>
          <p>{{mensagem.texto}}</p>
        </div>
      </div>
```

Figura 18 - *View chat.html do CIn Bot*

4.2 Utilização do serviço

O detalhamento de instalação e uso do serviço do Usain Bots está contida em sua documentação[13] e pode ser utilizada segundo as necessidades da aplicação cliente. Todo o código está disponível para futuras contribuições em:

[13] Documentação Usain Bots - http://docs.usainbots.apiary.io/

https://github.com/jorgehclinhares/usainbots

Será apresentado de maneira explícita critérios de uso do serviço por meio da correlação dos mesmos com as características encontradas na seção 3.1.1.

4.2.1 Identificação de abertura e cumprimento

Como proposto, a Figura 19 apresenta a aplicação que realiza sempre a abertura do diálogo e compreende uma cumprimento por meio do interlocutor instigando a ele sobre algum questionamento. Este critério atende noções de pró-atividade

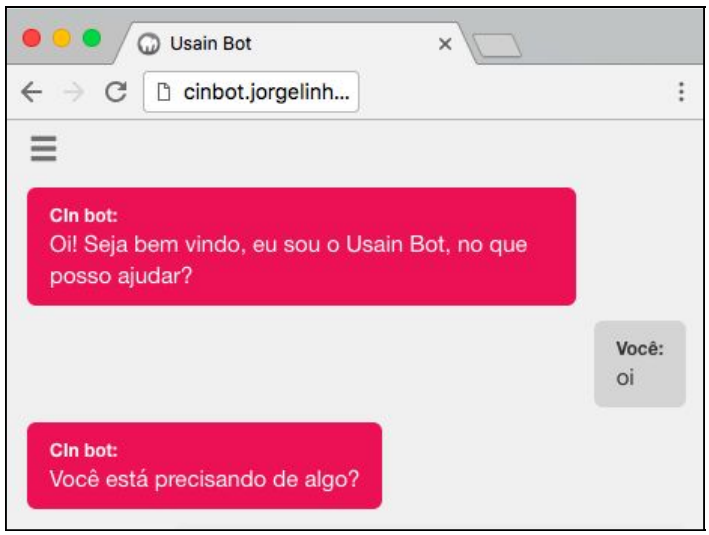

Figura 19 *- CIn bot: Identificação de abertura e cumprimento*

4.2.2 Identificação de questionamento e proposição de sugestões

A figura 20 e 21 demonstram a ação de identificação de questionamento sem mesmo a necessidade de uso da interrogação, o serviço do Usain Bots propõe a resposta esperada pelo interlocutor além de sugestões de novos temas ao qual ele detém conhecimento.

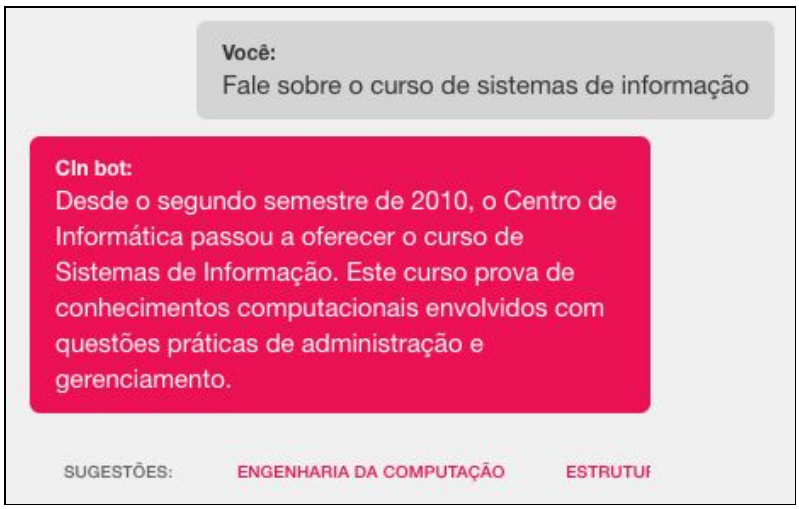

Figura 20 *- CIn bot: Identificação de questionamento*

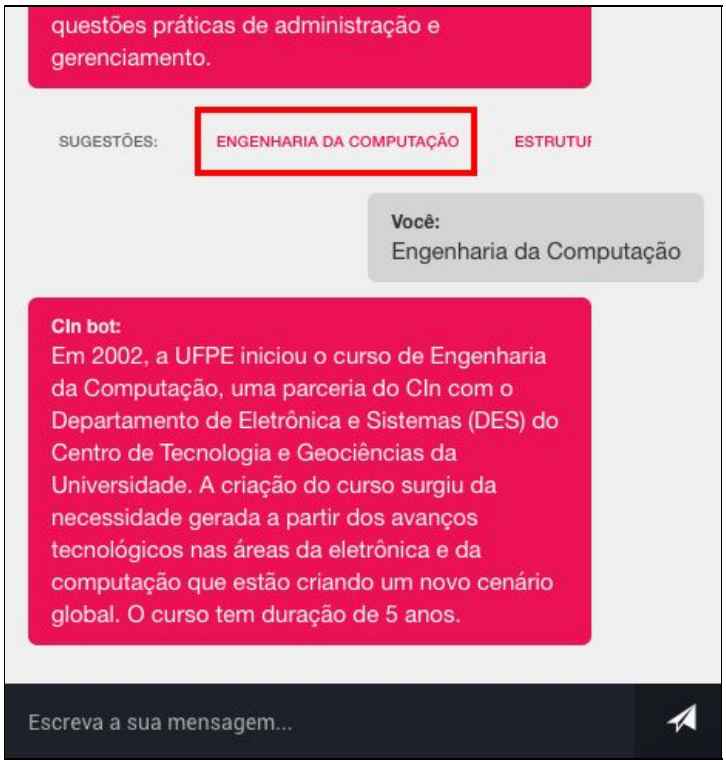

Figura 21 *- CIn bot: Ação de engajamento por sugestão*

4.2.3 Identificação de questionamento e sem resposta viável

Como exemplificada na Figura 22, o serviço do Usain Bots identificou um questionamento que não compreendia resultados esperados na sua base de conhecimento, então ele informou desconhecer sobre desconhecimento do assunto e também propôs novos assuntos como o visto na seção 5.3.2.

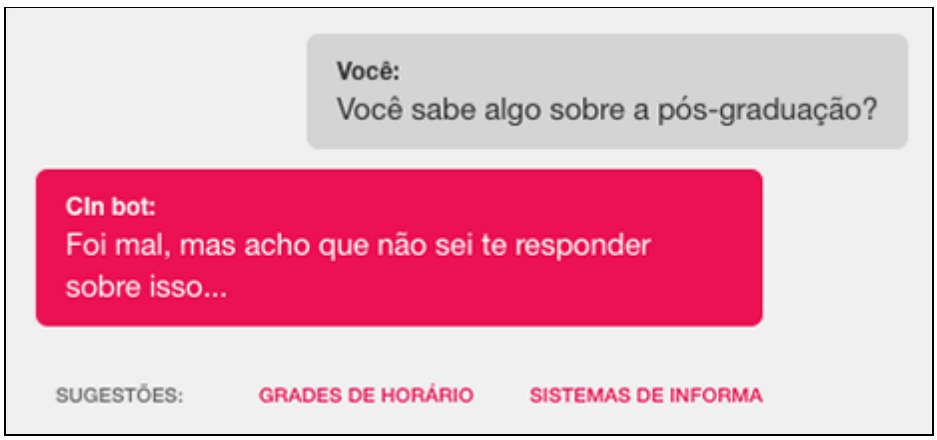

Figura 22 *- CIn bot: Identificação de questionamento sem resposta viável*

4.2.4 Identificação de agradecimento

É uma das funções do serviço Usain Bots está em identificar também agradecimentos por parte do interlocutor e assim retribuir de maneira igualitária o gesto proporcionado. É visto

na Figura 23 a representação do diálogo que acrescenta a resposta um engajamento em forma de questionamento sobre outros assuntos de interesse do interlocutor.

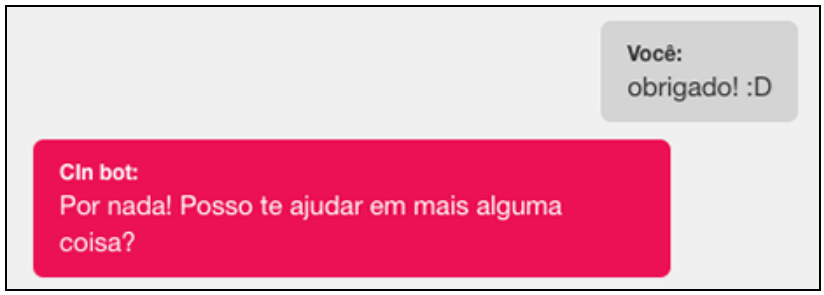

Figura 23 *- CIn bot: Identificação de agradecimento*

4.2.5 Funções extras

Aos serviços propostos pelo Usain Bots, foram acrescentados categorias de identificação específicas para o CIn Bot, são elas:

- **Notícia:** Diz respeito a identificação da intenção de uma busca pelas notícias mais recentes.

- **Evento:** Compreende a identificação da intenção de uma busca pelos eventos mais recentes.

Por meio da identificação destas categorias específicas que diferem das encontradas na seção 5.2, o CIn bot é capaz de coletar por meio da técnica de *web scraping* as últimas notícias e eventos relacionados ao Centro de Informática da UFPE, como por ser visto na Figura 24.

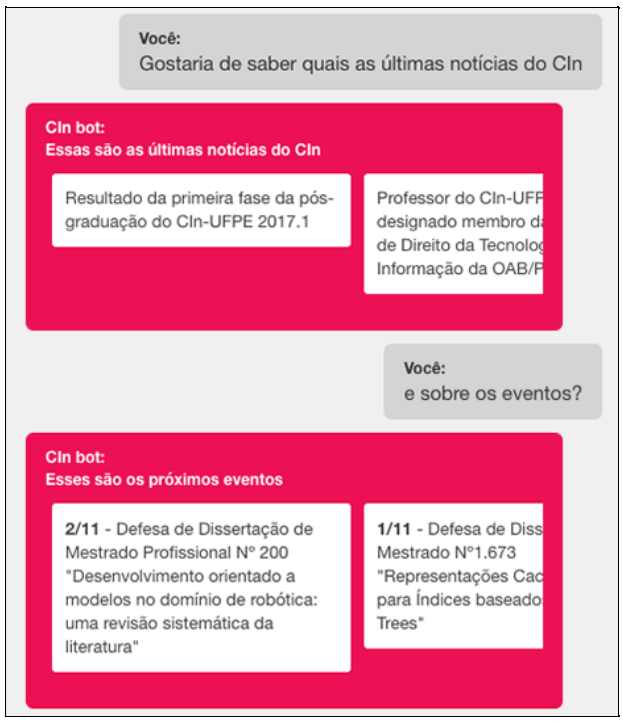

Figura 24 *- CIn bot: Identificação de notícias e eventos*

4.2.6 Identificação de encerramento

Como também mencionado na seção 5.2, o serviço do Usain Bots identifica o classificação encerramento como uma oportunidade de compreender que o interlocutor deseja encerrar o diálogo, sendo assim, é proposto uma despedida ao interlocutor como pode ser vista na Figura 25.

Figura 25 *- CIn bot: Identificação de encerramento do diálogo*

Capítulo 5

5. Conclusão e trabalhos futuros

Devido aos esforços e estudos oriundos da década de 50 a Inteligência artificial vem evoluindo e hoje atinge um patamar ao qual a projeta como uma tecnologia que pode trazer potencial de inovação para uma organização, a adoção de ferramentas de IA vem se tornando cada vez mais comum, uma destas ferramentas são os *chatterbots* ou *bots* como foram tratados neste trabalho. Estas ferramentas se mostraram verdadeiramente poderosas pelo seu potencial de interação direta com o ser humano, viabilizando um canal de comunicação que pode ser explorado por inúmeros contextos de uso.

O objetivo principal deste trabalho foi de compreender a essência e conceitos que levam a construção de um *bot*, a fim de propor um serviço web *open source* que auxilia nesta atividade. Desta forma, desenvolvedores podem centralizar os seus esforços unicamente para construir levar o processamento lógico de um *bot* para a nuvem, ocasionando em uma redução de esforços na criação de bots como um *software* centralizado.

O serviço Usain Bots proposto explorou as principais necessidades de um *bot*, assim como na compreensão das suas características fundamentais, da necessidade por aquisição de conhecimento e pelas regras de identificação de intenção. Além disso, outras características essenciais para serviço *web* como este, aplicando mecanismos de controle de acesso e gerenciamento de aplicações clientes.

As limitações do trabalho incluem uma melhoria no módulo de aquisição de conhecimento da *web* que se adapte a novas necessidades e que resulte em uma melhor precisão de coleta, a falta de uma dicionário de sinônimos e adoção das demais características de um *bot* em sua totalidade.

Trabalhos futuros incluem:

- Implementação da capacidade de memória sobre o diálogo, fazendo com que o *bot* possa recuperar informações que já ocorreram em um diálogo.

- Adição da possibilidade de conexão com outros *bots*, fazendo com que trocas de conhecimento específicos entre *bots* possam ser viabilizadas.

- Adição da característica de flexibilidade e capacidade de adaptação que podem resultar em uma maior eficiência de compreensão de um ambiente criado pelo interlocutor. Fazendo que o *bot* possa responder a estímulos que não foram previamente definidos.

- Adição da característica de carácter, fazendo com que o *bot* possa responder a estímulos de sentimentos e que possa apresentar a sua própria personalidade.

- Adoção de um dicionário de sinônimos aplicados sobre um questionamento do interlocutor para que ocorra uma melhor chance de recuperação de um documento no sistema de recuperação da informação.

Referências Bibliográficas

[1] BRACHMAN, Ronald J.; LEVESQUE, Hector J.. **Knowledge representation and reasoning:** subtítulo do livro. 1 ed. [S.L.]: MORGAN KAUFMANN, 2014. 413 p.

[2] CASTANHO, Carla Lisiane De Oliveira; WAZLAWICK, Raul Sidnei. **A avaliação do uso de chatterbots no ensino através de uma ferramenta de autoria**. Disponível em: <http://www.br-ie.org/pub/index.php/sbie/article/view/160>.Acesso em: 12 jan. 2012.

[3] COMARELLA, Rafaela Lunardi; CAFÉ, Lígia Maria Arruda. **Chatterbot: conceito, características, tipologia e construção**. Disponível em: <http://www.ies.ufpb.br/ojs/index.php/ies/article/view/1758>.Acesso em: 12 jan. 2012.

[4] DB-ENGINES. **Db-engines ranking**. Disponível em: <http://db-engines.com/en/ranking>. Acesso em: 12 dez. 2016.

[5] FILHO, Eustáquio Cézar Pereira. **O uso do processamento de linguagem natural na construção de chatterbots**. Disponível em: <https://dcc.catalao.ufg.br/up/498/o/Eustaquio2009.pdf>. Acesso em: 12 jan. 2012.

[6] GONZALEZ, Marco; LIMA, Vera L. S. De. **Recuperação de informação e processamento da linguagem natural**. Disponível em: <http://www.inf.pucrs.br/~gonzalez/docs/minicurso-jaia2003.pdf>. Acesso em: 12 jan. 2012.

[7] HARRINGTON, Peter. **Machine learning in action:** subtítulo do livro. 1 ed. [S.L.]: Manning, 2012. 382 p.

[8] JWT. **Introduction to json web tokens**. Disponível em: <https://jwt.io/introduction/>. Acesso em: 12 dez. 2016

[9] KAO, Anne; POTEET, Stephen R. **Natural language processing and text mining**. 1 ed. [S.L.]: Springer, 2007. 272 p.

[10] LAVEN, Simon. **The Simon Laven Page**. Disponível em: <http://www. simonlaven.com>. Acesso em: 12 dez. 2016.

[11] LEONHARDT, Michelle Denise. **Doroty: um chatterbot para treinamento de profissionais atuantes no gerenciamento de redes de computadores**. Disponível em: <http://www.lume.ufrgs.br/handle/10183/5659?locale=pt_br>.Acesso em: 12 jan. 2012.

[12] PEREIRA, Caio Ribeiro. **Node.js - aplicações web real-time com node.js:** Aplicações web real-time com Node.js. 1 ed. [S.L.]: Casa do Código, 2014. 158 p.

[13] PEREIRA, Silvio Do Lago. **Raciocínio automatizado**. Disponível em: <https://www.ime.usp.br/~slago/ia-raciocinioautomatizado.pdf>.Acesso em: 12 dez. 2016.

[14] ROTHERMEL, Alessandra; DOMINGUES, Maria José Carvalho De Souza. **Maria: um chatterbot desenvolvido para os estudantes da disciplina "métodos e técnicas de pesquisa em administração"** . Disponível em: <http://www.aedb.br/seget/arquivos/artigos07/923_artigos2007eget2.pdf>.Acesso em: 12 jan. 2012.

[15] RUSSEL, Stuart; NOVING, Peter. **Inteligência artificial.**. 3 ed. [S.L.]: Campus, 2013. 1324 p.

[16] SHAWAR, Bayan Abu; ATWELL, Eric; ROBERTS, Andrew. **Faqchat as an information retrieval system.** Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.63.8564>.Acesso em: 12 jan. 2012.

[17] SOUCI, Benjamin San; LEMAIRE, Maude. **An inside look at the architecture of nodejs**. Disponível em: <http://mcgill-csus.github.io/student_projects/submission2.pdf>.Acesso em: 12 jan. 2012.

[18] TURING, Alan Mathison. **Computing Machinery and Intelligence**. Disponível em: <https://www.csee.umbc.edu/courses/471/papers/turing.pdf/>. Acesso em: 20 nov. 2016.